AD_________________

AWARD NUMBER: W81XWH-07-1-0483

TITLE: In Silico Genome Mismatch Scanning to Map Breast Cancer Genes in Extended Pedigrees

PRINCIPAL INVESTIGATOR: Alun Thomas, Ph.D.

CONTRACTING ORGANIZATION: University of Utah
Salt Lake City, UT 84112

REPORT DATE: July 2009

TYPE OF REPORT: Annual

PREPARED FOR: U.S. Army Medical Research and Materiel Command
Fort Detrick, Maryland 21702-5012

DISTRIBUTION STATEMENT: Approved for Public Release;
Distribution Unlimited

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision unless so designated by other documentation.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE 1 July 2009	**2. REPORT TYPE** Annual	**3. DATES COVERED** 15 Jun 2008 – 14 Jun 2009

4. TITLE AND SUBTITLE In Silico Genome Mismatch Scanning to Map Breast Cancer Genes in Extended Pedigrees	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** W81XWH-07-1-0483 **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) Alun Thomas, Ph.D. E-Mail: alun.thomas@utah.edu	**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Utah Salt Lake City, UT 84112	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Medical Research and Materiel Command Fort Detrick, Maryland 21702-5012	**10. SPONSOR/MONITOR'S ACRONYM(S)** **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This project aims to map breast cancer genes using dense single nucleotide polymorphism assays in large extended pedigrees. Data has been collected using 1,000,000 SNP genotype assays for 25 women affected by breast cancer in three high risk Utah pedigrees. Analysis of control data from the HapMap project has been completed and methods that will model linkage disequilibrium for genome wide, dense, SNP data have been developed.Papers describing these methods have been published. Programs implementing these methods have beenwritten, tested and released. The programs are now being applied to the Utah breast cancer susceptibility families and we expect the results to be submitted for publication by the new end of this project.

15. SUBJECT TERMS
Shared genomic regions, linkage disequilibrium modeling, pedigree analysis, single nucleotide polymorphisms

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** USAMRMC
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	UU	89	**19b. TELEPHONE NUMBER** *(include area code)*

Contents

Introduction

The purpose of this project is to exploit high density single nucleotide polymorphism (SNP) assays to map genes for breast cancer in clusters of cases related through large extended pedigrees. The central idea is to search for long runs of markers where cases share a common allele. Unusually long runs indicate regions where the cases share a segment of chromosome identical by descent from a common ancestor. If sharing of such a segment is sufficiently rare by chance, the segment becomes a candidate as a region containing a gene for breast cancer. The probability that a random segment reaches or exceeds the length of the longest observed shared segment can be assessed by simulation. One of the major challenges in this project is to properly account for (LD), that is, the fact that in high density marker panels the alleles at nearby markers are correlated. Conventional methods generally assume no correlation between markers, however, this will lead to improper assessment of the statistical significance of the observed shared regions. As well as analyzing the high density data collected under this project, we expect the methods and programs we develop to be applicable in similar study designs for other diseases.

Body

Aim 1: collection of data

Our first aim was to obtain genome wide SNP data for selected cases in three high risk breast cancer pedigrees. This was done in the first year of the project and reported in our previous annual report. Note that although we originally proposed an assay of 550,000 loci for each sample, by the time the data was collected the standard assay had more than 1,000,000 SNPs. This continues to be the current standard. We also obtained suitable control data from the HapMap project.

Aim 2: statistical developments

The major statistical element in this project was to develop models to account for LD between SNP markers in genome wide assays. As the density of the standard assay increased this became both more timely and more challenging. However, we have made considerable progress. The original approach to scaling up the number of loci that could be handled was to restrict the graphical models used to model LD to those that have conditional independence graphs that are interval graphs. This was described by Thomas (2009*b*), a paper that was in press at the time of our previous report, but which has now appeared. The implementing program accompanying this paper could, at that time, handle up to 20,000 loci simultaneously. We have since made changes to the method that makes a further restriction limiting the maximum possible extent of LD around a locus. This emphasizes the tendency of interval graphs to be linear, or long and thin, in structure and enabled us to use a walking window approach in the model estimation phase. This greatly increased the number of loci that can be handled: our current implementation has been used on data sets with over 200,000 loci which is far more than the number of loci that are usually assayed on chromosome 1, the longest chromosome. Since the chromosomes can be handled independently, this achieves our goal of practical and efficient LD modeling on a genome wide scale. The methodology behind this development is described in Thomas (2009*c*) and an example of its use in the analysis of a high risk prostate cancer family is described in Thomas (2009*a*).

One very useful application of our graphical models for LD is to realistically simulate complete chromosome haplotypes and genotypes from the models. Such data can be used to develop statistical methods, assess the significance of test statistics using empirical p-values, and to compute statistical power. We have developed programs that take the output from our model estimation programs and produce such simulations. These can be used to simulate a population sample of unrelated individuals, and to simulate data from relatives given an extended pedigree structure. Pedigree data is simulated using a modified form of the gene-drop simulation method (MacCluer et al. 1986) which assigns founder haplotypes from an LD model and then simulates their descent to other family members. This methodology and an example are also described in Thomas (2009*c*) and Thomas (2009*a*) respectively. This approach is used in the data analysis of our breast cancer pedigrees as described below.

We have realized that in addition to searching for segments where relatives share genome identical by descent heterozygously, it is also informative to look for regions of homozygous sharing. This can be applied to both relatives from a known pedigree and to individuals from a founder population. It can indicate regions where there may be either genomic deletions or a region containing a recessive locus. This idea has previously been suggested by Miyazawa et al. (2007) but their statistical assessment of any sharing was done using asymptotic distribution theory whose assumptions may not be matched in a real data analysis. We have now extended our LD modeling and gene drop simulation approach to such homozygous sharing.

Finally, the application of graphical model estimation to LD has stimulated some theoretical work that is applicable to graphical model estimation in a general context. This has been done in collaboration with Professor Peter Green at the University of Bristol. Thomas & Green (2009*b*) gives a method for enumerating the number of equivalent junction tree representations that a graphical model can have, while Thomas & Green (2009*a*) enumerates the decomposable graphs that can be obtained by either adding or deleting an edge from a given decomposable graph. While these results may seem somewhat esoteric, they have real applications to the problem of optimizing a decomposable graphical model by using Markov chain Monte Carlo methods, an approach that is ubiquitous in this field. We are currently working on an implementation of a new class of MCMC optimization scheme that exploits these results.

Aim 3: software development

All of the methods described above have been implemented in Java programs that are available from the PI's web site (www-genepi.med.utah.edu/∼alun/software). These run on Windows, Max, Unix and Linux environments. All the programs use the existing standard LINKAGE program format for input and output of genetic data (linkage.rockefeller.edu), which is familiar to most users. Included in the distribution are compiled classes, source code, example data sets, and documentation comprising web pages generated using the `javadocs` program. The programs are:

- `SGS` This is new name for the prototype program we called `Shags` in our previous annual report. It finds genomic segments shared heterozygously in a set of individuals who can be unrelated or related by a known extended pedigree. The program has been run to analyze data for over 200,000 loci in a single run.

- `SimSGS` This is the new name for the prototype program we called `SimShags` in our previous annual report. This simulates data to match that analyzed using `SGS` using a multi locus gene drop approach, and hence obtains the distribution of maximum genome sharing statistics under the null hypothesis of no genetic effect. It allows simulation of founder haplotypes under either linkage equilibrium or LD. The program has been run to simulate data for over 200,000 loci in a single run.

- HGS This is used in a similar way to SGS but computes runs of homozygous, rather than heterozygous, sharing.

- SimHGS This is used in a similar way to SimSGS but computes the null distribution of statistics for runs of homozygous sharing.

- IntervalLD This is the new name for the prototype program we called IntervalHapGraph in our previous annual report. This searches for an optimal graphical model for LD structure given a sample of unrelated individuals from the sampled population. It restricts the class of models to those with interval conditional independence graphs and implements a walking window approach in the search. Its computational requirements, time and storage, have been shown to increase linearly with both the number of loci considered and the number of individuals in the sample. The output from this program provides the input LD model for SimSGS and SimHGS. This program has also been run and tested on data sets with over 200,000 loci.

- GeneDrops This is a new program that simulates genotypes in a pedigree under either linkage equilibrium or LD to match the data in a given input pedigree. It will produce multiple simulations from a single call. The simulation method used is the same as that used by SimSGS and SimHGS; however, in this case the complete simulated pedigrees are output, not just summary statistics, thus, it can be employed to evaluate arbitrary statistical analyses.

Aim 4: data analysis and publication

The analysis of our three breast cancer susceptibility families is proceeding. The introduction of the 1,000,000 SNP Illumina assay took slightly longer to acquire than we anticipated and the larger data set required some modifications in our developments in order to handle this scale up. Hence, the analysis is not yet complete and in view of this we were allowed a no cost extension to conclude it and prepare results for publication.

To date we have completed an analysis of the control data obtained from the HapMap project and prepared a manuscript for publication (Cai et al. 2009) describing, and explaining, some anomalous features that could have let to false positive results if left uninvestigated. This manuscript is in preparation.

We have obtained a profile of long shared genomic segments throughout the genome for the affected individuals in each of our three sampled pedigrees using SGS. We have also estimated graphical models for LD from the control data using IntervalLD and obtained simulations of genomic sharing under the null hypothesis of no genetic effect for all our pedigrees using SimSGS. At present we are tuning the parameters of the programs in order to get the most reliable estimates and realistic simulations. We expect this work to be completed and submitted for publication before the end of the no cost year.

Key research accomplishments

- Development of methods that can estimate graphical models for LD on a genome wide scale for high density SNP assays.

- Development of simulation methods for high density SNP assay data under LD in pedigrees or unrelated individuals.

- Release of software, including source code and documentation, implementing the above methods.

- Analysis of control data from the HapMap project indicating problem areas of the genome which will allow us to avoid false positive results.

- Initial analysis of 1,000,000 SNP markers on 25 breast cancer cases in three high risk pedigrees.

Reportable outcomes

- Thomas (2009*b*) published.
- Thomas (2009*c*) published.
- Thomas (2009*a*) published.
- Thomas & Green (2009*a*) published.
- Thomas & Green (2009*b*) in press.
- Cai et al. (2009) submitted for publication.
- Poster presentation based on Cai et al. (2009) made at International Genetic Epidemiology Society meeting, St Louis, September 2008.
- Seminar based on Thomas (2009*c*) and Thomas (2009*a*) presented at Deutsches Krebsforschungszentrum, Heidelberg, Germany. May 2009.
- Seminar based on Thomas (2009*c*) and Thomas (2009*a*) presented at the Faculty of Agricultural Sciences, Aarhus University, Foulum, Denmark. May 2009.

Conclusion

Aim 1, collection of data; aim 2, statistical developments; and aim 3, software development, are now complete. Papers describing the methods have been published and programs implementing them have been developed, tested, and released. We have also made significant progress under aim 4, data analysis and publication. Analysis of control data has been completed and submitted for publication. Analysis of case data is well underway and should be submitted for publication before the end of the project.

References

Cai, Z., Allen-Brady, K. & Thomas, A. (2009), Anomalous shared genomic segments in high risk cancer pedigrees and hapmap control data. Submitted.

MacCluer, J. W., Vandeburg, J. L., Read, B. & Ryder, O. A. (1986), Pedigree analysis by computer simulation, *Zoo Biology* **5**, 147–160.

Miyazawa, H., Kato, M., Awata, T., Khoda, M., Iwasa, H., Koyama, N., Tanaka, T., Huqun, Kyo, S., Okazaki, Y. & Hagiwara, K. (2007), Homozygosity haplotype allows a genomewide search for the autosomal segments shared among patients, *American Journal of Human Genetics* **80**, 1090–1102.

Thomas, A. (2009*a*), Assessment of SNP streak statistics using gene drop simulation with linkage disequilibrium, *Genetic Epidemiology.* In press.

Thomas, A. (2009*b*), Estimation of graphical models whose conditional independence graphs are interval graphs and its application to modeling linkage disequilibrium, *Computational Statistics and Data Analysis* **53**, 1818–1828.

Thomas, A. (2009*c*), A method and program for estimating graphical models for linkage disequilibrium that scale linearly with the number of loci, and their application to gene drop simulation, *Bioinformatics* **25**, 1287–1292.

Thomas, A. & Green, P. J. (2009*a*), Enumerating the decomposable neighbours of a decomposable graph under a simple perturbation scheme, *Computational Statistics and Data Analysis* **53**, 1232–1238.

Thomas, A. & Green, P. J. (2009*b*), Enumerating the junction trees of a decomposable graph, *Journal of Compuational and Graphical Statistics.* In press.

Appendices

BIOINFORMATICS

Vol. 00 no. 00 2008
Pages 1–6

A method and program for estimating graphical models for linkage disequilibrium that scale linearly with the number of loci, and their application to gene drop simulation

Alun Thomas

Department of Biomedical Informatics, University of Utah.
Genetic Epidemiology, 391 Chipeta Way Suite D, Salt Lake City, UT 84108, USA

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Motivation: Efficient models for genetic linkage disequilibrium are needed to enable appropriate statistical analysis of the dense, genome wide single nucleotide polymorphism assays currently available.

Results: Estimation of graphical models for linkage disequilibrium within a restricted class of decomposable models is shown to be possible using computer time and storage that scale linearly with the number of loci. Programs for estimation and for simulating from these models on a whole genome basis are described and provided.

Availability: Java classes and source code for `IntervalLD` and `GeneDrops` are freely available over the internet at `http://bioinformatics.med.utah.edu/∼alun`.

Contact: alun@genepi.med.utah.edu

1 INTRODUCTION

Linkage disequilibrium, or *LD*, is the non-independence of alleles at proximal genetic when the recombination process along chromosomes has not had enough time to randomize their states. It is a form of *allelic association* other forms of which can also arise due to selection, relatedness of individuals, and population admixture. Statistical methods that do not take LD into account can be misled into false results. In sparse sets of genetic loci, its effects can be negligible, however, with the dense genetic assays using single nucleotide polymorphisms, or *SNPs*, currently available, it is critical to properly account for it. Several approaches have been employed for modelling LD, in particular graphical modelling has been used extensively in this context. Verzilli *et al.*, 2006 estimated graphical models for correlated genotypes at proximal loci, while Scheet and Stephens, 2006 defined graphical models on variables indicating the cluster of origin of alleles and genotypes. Thomas and Camp, 2004 , Thomas, 2005 , Thomas, 2007 , and Thomas, 2009 developed methods for estimating graphical models in which the variables are the alleles themselves, and it is this development that we continue here. The estimation approach presented here is similar to that of Greenspan and Geiger, 2004 . Their use of the EM algorithm in addressing the missing data problem of unknown phase is similar to the two phase approach described below which may be thought of as a stochastic version of EM where the E-step uses a random imputation instead of a conditional mean. The classes of graphical models used are, however, quite different. Where Greenspan and Geiger, 2004 use a Bayesian network to explicitly model the haplotype block structure of the genome, this work models the data in a purely empirical manner using the physical relationships between the loci only to restrict the class of decomposable graphical models that can be considered.

A graphical model for a set of random variables is based on a factorization of their joint distribution as

$$P(X_1, \dots, X_n) = \prod_i f_i(T_i)$$

where $T_i \subset \{X_1, \dots, X_n\}$, and each f_i is some non-negative function of a subset T_i of the variables. From this we define a graph in which each variable is a vertex, and pairs of vertices are connected by edges if they are both contained in the same T_i. This is called the *conditional independence graph* or *Markov graph* as it allows the conditional independence relationships between the variables defined by the factorization to be easily read off.

Højsgaard and Thiesson, 1995 originally developed methods for estimating graphical models for discrete random variables by maximizing a penalized likelihood function. The penalty, based on the number of parameters, is necessary to avoid fully saturated models. Giudici and Green, 1999 developed estimation of graphical models on Gaussian variables as did Dobra *et al.*, 2003 and Jones *et al.*, 2005 . Thomas and Camp, 2004 adapted the method of Højsgaard and Thiesson, 1995 for estimating graphical models for LD from phase known haplotype data, replacing the original deterministic model search with the random methods of Metropolis sampling Metropolis *et al.* (1953) and simulated annealing Kirkpatrick *et al.* (1982). Thomas, 2005 extended this approach to use unphased genotype data using a two stage method. Given an initial model, usually the trivial one that represents linkage equilibrium, and the observed genotypes, complete phase known haplotypes are imputed for the sampled individuals. Then, given these imputed haplotypes, the graphical model is re-estimated. Given a new graphical model, the haplotypes are re-imputed, and so on.

The above methods restrict the search for graphical models to those whose conditional independence graphs are *decomposable* or, equivalently, *chordal* or *triangulated* Golumbic (1980). It is this property that allows the likelihood and degrees of freedom of a graphical model to be computed. Decomposable graphs, however, are not easily characterized. The random search methods described above typically propose changes to an incumbent graph such as adding or deleting an edge. Before evaluating the penalized likelihood for the proposed graph, it is necessary to first check that it is decomposable. While Giudici and Green, 1999 give efficient methods for this, in large graphs the probability that a random proposal will be decomposable decreases rapidly, ultimately making the search procedure very inefficient. Thomas, 2009 circumvented this problem by restricting the search to the easily characterized subclass of models whose conditional independence graphs are *interval graphs*, as defined below. This restriction was shown to greatly improve the search efficiency, without sacrificing power to appropriately model LD.

In this work we add one further model restriction that enables a walking window approach to estimation of LD between the alleles along a chromosome. This is implemented in a program whose storage and time requirements are linear in the number of loci considered as we illustrate on examples of over 200000 markers taken from the HapMap YRI data set The International HapMap Consortium (2005).

While Thomas, 2007 showed that graphical models for LD could, in principle, be used for linkage analysis, full multi point linkage analysis is not feasible with dense SNP data. However, it is feasible to use haplotypes generated using LD models in simulation methods to assess significance in association studies involving unrelated cases and controls. It is also feasible to generate haplotypes for pedigree founders using the models and then simulate the descent of the alleles to descendants using the multi locus gene drop method. These simulations can be used for assessing the statistical significance of long runs of allele sharing that are used in the method of mapping by shared genomic segments introduced by of Thomas *et al.*, 2008 and Leibon *et al.*, 2008 . With this in mind, we have written a program to perform gene drop simulation with linked markers in LD.

In what follows we briefly review estimating graphical models from genotype data and describe the role of interval graph in this context. We then describe the implementation of this approach in a walking window along the chromosome. We also describe briefly how the programs implementing these methods can be used. Finally, we illustrate the linearly scaling performance of the program with data from HapMap.

2 METHODS

2.1 Estimating graphical models

A graphical model is decomposable if and only if the maximal cliques, $C_1, \dots C_c$, of its conditional independence graph can be ordered so that the following *running intersection property* holds:

$$S_i = C_i \cap \bigcup_{j=i+1}^{c} C_j \quad \subset C_k \ \text{ for some } \ k > i.$$

The sets S_i are called the *separators* of the graph, by convention $S_c = \emptyset$. The joint probability distribution can then be expressed as a function of the clique and separator marginals:

$$P(X_1, \dots, X_n) = \prod_i \frac{P(C_i)}{P(S_i)}.$$

This then allows the calculation of the maximized log likelihood and degrees of freedom for the graphical model as

$$\log(\hat{L}(G)) = \sum_i \log(\hat{L}(C_i)) - \sum_i \log(\hat{L}(S_i))$$

and

$$\text{df}(G) = \sum_i \text{df}(C_i) - \sum_i \text{df}(S_i)$$

respectively, from which we can obtain the penalized likelihood or *information criterion*

$$IC(G) = \log(\hat{L}(G)) - \alpha\, \text{df}(G). \qquad (1)$$

In the case of discrete data with no missing values, the clique and separator marginals are simply contingency tables for which the degrees of freedom and maximized likelihood are easily calculated.

As noted above, we can optimize $IC(G)$ using random search methods in which we make small changes to G to obtain G' which then must be checked for decomposability and subject to the usual Metropolis or simulated annealing rejection step.

In order to estimate graphical models from genotype data, we must first impute complete phase know data under an initial model. For this we assume linkage equilibrium which is equivalent to a graph G with no edges. The above random search method is then run for some number of iterations and the resulting graphical model and maximum likelihood parameter estimates are used to obtain new imputations for the complete phase known data. Details of this process are given by Thomas, 2005 .

2.2 Interval graphs

Under a perturbation scheme that simply adds or deletes edges from the incumbent graph, as the number of vertices increases the probability that a random proposal G' is decomposable decreases. Thomas, 2009 shows that for the LD problem, the probability decreases approximately as $1/n$ but that this can be avoided by restricting the conditional independence graphs to be *interval graphs*. A graph is an interval graph if and only if the vertices can be made to correspond to intervals of the real line such that two vertices are connected if and only if their corresponding intervals overlap. This has some intuitive appeal for the LD problem because loci are ordered linearly along a chromosome, and we expect that LD will decay with distance. In order to reflect this, we assign each locus a point on the line and require its corresponding interval cover its location. In this application the points are evenly spaced in chromosomal order, but could be made to reflect the physical distances between loci. Thomas, 2009 also required that in order for two vertices to be connected, their intervals must overlap by a minimum, non-zero amount. This allows some flexibility for loci positioned between two correlated loci, but which appears to be stochastically independent, to be assigned a small interval and hence avoid a forced connection with one of the flanking loci.

Interval graphs are a sub class of decomposable graphs Golumbic (1980), and it is easily shown that the additional restriction described above still define interval graphs. Moreover, the characterization of the graph structure in terms of intervals of the line make it simple to propose new graphs in the same class without having to check for decomposability. For example, if we propose changing the length of an interval the resulting perturbed graph is obviously still an interval graph, and hence decomposable. Thomas, 2009 showed that this leads to considerable computational efficiencies, and that the haplotype frequencies from models with interval graphs do not differ greatly from those under general decomposable models, nor from those implied by the models of Scheet and Stephens, 2006 .

In order to store and manipulate the intervals, Thomas, 2009 used a standard structure called an *interval tree* de Berg *et al.* (2000). This structure

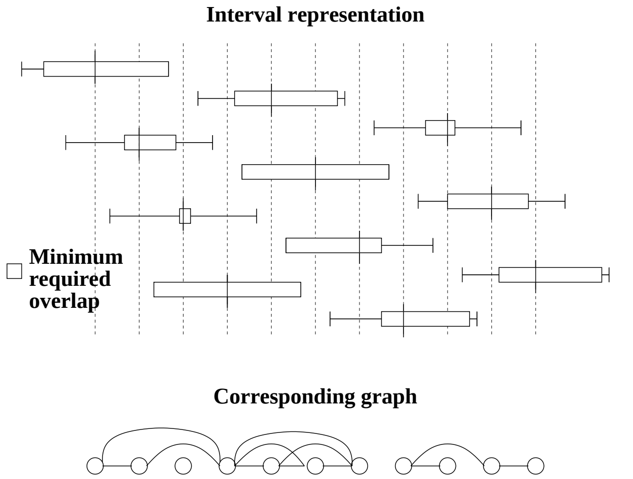


Fig. 1. The relationship between the interval representation of an interval graph and the graph itself. A box represents the interval assigned to a locus which is constrained to include the locus's assigned position, shown as a line crossing the box. The whiskers represent the maximum extent allowed for the interval. Boxes that overlap by at least the minimum requirement lead to connections between the vertices in the corresponding graph, shown below. Note that the interval corresponding to locus 3 is shorter than the minimum required overlap, shown on the left hand side, and hence the vertex in the corresponding graph has no connections. Similarly, the overlap between the intervals for loci 9 and 10 is insufficient to create a link. Note also that the graph shown here and the one in figure 2 are both decomposable which is a consequence of their derivation from the interval representation.

allows addition and deletion of intervals and queries as to which intervals overlap with a given one to be carried out in, at best, $O(\log(n))$ time. Together with the the required storage manipulations this resulted in super linear time requirements for large data sets of 10,000 loci or more. To overcome this, we now introduce a final model restriction: we require that the interval representing a locus extends no more that some maximum value μ to each side of its associated point. This allows the intervals to be stored in a simple array ordered by the position of the required point. To identify the intersectors of an interval we now need only find its index in the array and check each interval whose required point is within 2μ units each side. Such a query can be done in time independent of the size of the array. The structure of these graphs is illustrated in figure 1. While this makes graph updates very efficient, imputing the phase known haplotypes for all loci is still a computationally demanding step. For this reason we have adopted the following walking window approach.

2.3 Graph updates in a window

The first stage of the search method involves searching the space of interval graphs given fixed, imputed haplotypes. In doing this we restrict the vertices being considered to those within a contiguous window of the line. We propose an update to the incumbent graph, G say, by choosing a vertex in the window and perturbing its corresponding interval by generating new random end points each side of the fixed point. The new distances to the end points are generated independently from the Uniform$(0, \mu)$ distribution. The proposed graph, G', is then either accepted as the next incumbent, or rejected, based on the value of the information criterion. Thomas, 2009 showed that both the likelihood and the degrees of freedom of G' can be evaluated by considering the values for G and the small subgraph within the maximum extent of the interval being updated. This is independent of the number of loci and hence very fast.

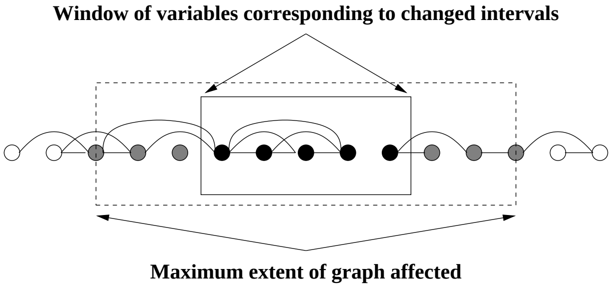

Window of variables corresponding to changed intervals
Maximum extent of graph affected

Fig. 2. The vertices possibly affected by updates to the graph in a fixed window. The intervals for the vertices in the window, the black circles in the solid box, can be changed by generating new end points. Edges between these and the grey vertices in the dotted box may change as a result, and these need to be considered in the new calculation of maximized likelihood and degrees of freedom. No part of the graph outside the dotted box can change until the window moves.

Although we limit the intervals updated to a fixed window, the effects can extend to either side of the window. This is illustrated in figure 2. In the example shown, a window of 5 variables is being updated, however, as figure 1 shows, the maximum length of an interval is 3.4 units, where a unit is the distance between two consecutive markers. Hence, a vertex may be connected to the any of the three previous or next vertices. The effects of changes to the window of 5 vertices may, therefore, extend to an enclosing window of 11. However, outside of this, no edges can change. In the implementation described below, the walking window covers 100 vertices at a time and an interval can extend be up to 8 units long.

2.4 Phase imputations in a window

The second stage of the search is to update the imputed haplotypes given the current model and the observed genotypes. Again we do this in a restricted contiguous window of loci. The current graphical model is applied to both the paternal and maternal haplotypes for each observed individual. The values at each locus determine the observed genotypes. Thus, we obtain a compound graphical model connecting haplotypes to genotypes as shown in figure 3. A random imputation for the haplotypes can now be made using the usual forward-backward graphical modelling methods as described by Thomas, 2009 . This requires determining from the graph an ordering of the variables to be updated. The forward step then proceeds through this list calculating the conditional distribution of the state of each given the states of those that appear later in the list. The conditional independences implied by the graph mean that this conditional distribution typically depends on only a small subset of the remaining variables so that this is usually a quick computation. The backward step then moves through the same list in reverse order using the conditional distributions previously calculated to simulate a state for each variable given the states of those already determined.

Note that in making these updates, we change only values in the current window, however, these may depend on the values of alleles at loci neighbouring the window, as shown in figure 3.

While the graphical models for the haplotypes, figure 2, are guaranteed by the interval graph representation to be decomposable, the compound graphical models, figure 3, are typically not. Consequently, to find the ordering of the vertices needed to make the forward-backward steps described above, we first need to find a triangulation of the graph within the window, and hence find a decomposition. This step is super linear in the time required, and this is why we implement the walking window approach. Initial testing showed that the computational time required to make this update grows approximately as $w \log(w)$ where w is the length of the window.

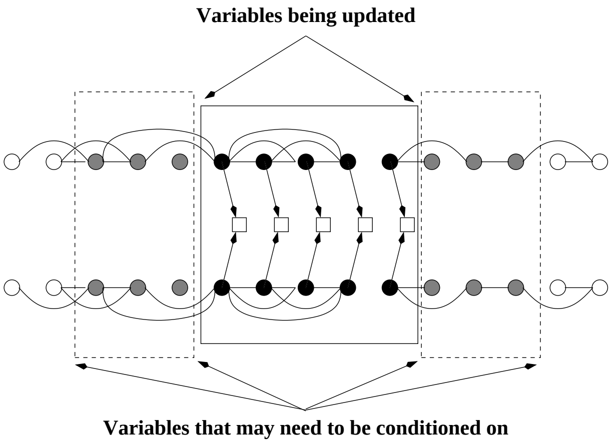


Fig. 3. The graphical model shown in figure 2 is applied to the paternal and maternal haplotypes of each individual in the sample in parallel. These affect the observed genotypes shown as white squares. The states of the black vertices are updated conditional on the genotypes and the current graphical mode. This may require conditioning on states of vertices outside the window, as shown in grey.

3 IMPLEMENTATION

3.1 Model estimation

The above methods have been implemented in a program called IntervalLD that is available as part of the author's packages of Java programs for graphical modelling and genetic analysis. The program takes input in the same standard file formats as the much used *LINKAGE* programs (http://linkage.rockefeller.edu). Two files are input: one specifying the nature of the genetic loci being analyzed, the second giving a list of individuals and their genotypes. The format allows for specifying pedigree relationships between the individuals, but also, by listing the parents as zero, allows for samples of unrelated individuals as required by these methods. IntervalLD will treat all individuals as unrelated even if relationships are specified.

The program is run using the following call

```
java IntervalLD in.par in.ped [w] [p] [g]
```

where

- in.par and in.ped are the LINKAGE format input files described above,
- w is the width of the window of loci to be considered. The default is 100.
- p is the number of phase updates to make in each window. The default is 5.
- g is the number of sweeps of graph updates to try between each phase update. The default is again 5.

In the above and what follows, arguments in square brackets – [] – are optional.

Having read in the data and set up the appropriate data structures, the program makes an initial imputation of phase based on the assumption of linkage equilibrium.

Then, one round of random sampling is made as follows. In each window, g sweeps are made of the loci in order, randomly perturbing the corresponding interval by proposing new end points, calculating the likelihood and degrees of freedom of the new graph, and either accepting or rejecting the change based on Metropolis acceptance probabilities. After g sweeps are made, a new phase imputation is made in the window. This new imputation is randomly chosen given the updated graphical model and the observed genotypes. This is repeated p times, with g Metropolis sweeps between each imputation. The window is then advanced along the chromosome by one half window length and the above process is repeated until the end is reached.

Following the round of random sampling a round of random uphill optimization is made. This follows the same format as the random sampling, but the Metropolis sampling of the graph is replaced by an uphill search: the proposed graph is accepted only if it is as good as or better than the current. Also, instead of making random phase imputations, imputations are made by choosing the most probable haplotypes. As with the random version, this is done using standard graphical modelling forward-backward methods.

The multinomial parameters of the resulting graphical model are then output to a file. The file format is straightforward and human readable, although it is intended primarily for input into other programs. It is a list of the conditional distributions of the state of the alleles at each locus given the values at the loci that the graphical model defines as relevant.

3.2 Gene drop simulation under LD

Gene drop is a method for simulating the genotypes of related individuals in a pedigree. Alleles are allocated to founders at random, and these are then dropped down the pedigree mimicking Mendelian inheritance until the genotypes of all individuals are allocated. The single locus version is described by MacCluer *et al.*, 1986 . The multi locus version is similar, the difference being that the probabilities of inheritances at successive loci depend on the recombination fraction between them. An implementation of multi locus gene drop is given in the *MERLIN* program Abecasis *et al.* (2001). The implementation given here differs only in that the founder alleles are allocated by simulating haplotypes from a graphical model as estimated from control data using IntervalLD. The program is called as follows:

```
java GeneDrops in.par in.ped n pfx [ldmod] [-a]
```

where

- in.par and in.ped are again LINKAGE format input files. In this case the pedigree structure specified in in.ped is used for simulations.
- n is the number of simulations to perform.
- pfx is a prefix used in naming the output files. For example if n is 10 and pfx is gdout then the output files will be named gdout.01, gdout.02, ... gdout.10. Each of these files will be a LINKAGE pedigree files differing from in.ped only in that the genotypes given by them are simulations rather than the actual observations.
- ldmod is the name of the file containing the graphical model for LD. Note that this can be omitted in which case standard gene drop simulations are made under the assumption of linkage equilibrium using the alleles frequencies given in in.par.
- -a is an optional argument that determines what to output. By default genotypes are only output to match those observed, as specified in the in.ped file: that is, if a genotype is unspecified in the input file it will be unspecified in the output file also. If the -a option is given, however, complete genotypes are output for all individuals.

4 RESULTS

To illustrate the linear scaling of IntervalLD and GeneDrops we ran both programs on SNP data for chromosome 1 downloaded from HapMap. We used the YRI data which are the genotypes of 30 parent-offspring trios of Yoruba people from Ibadan, Nigeria. For model estimation we used only data on the 60 unrelated parents. This first required rewriting the downloaded files to make LINKAGE format input files. We then selected subsets of different numbers of loci, the smallest being 100 the largest being all 223110

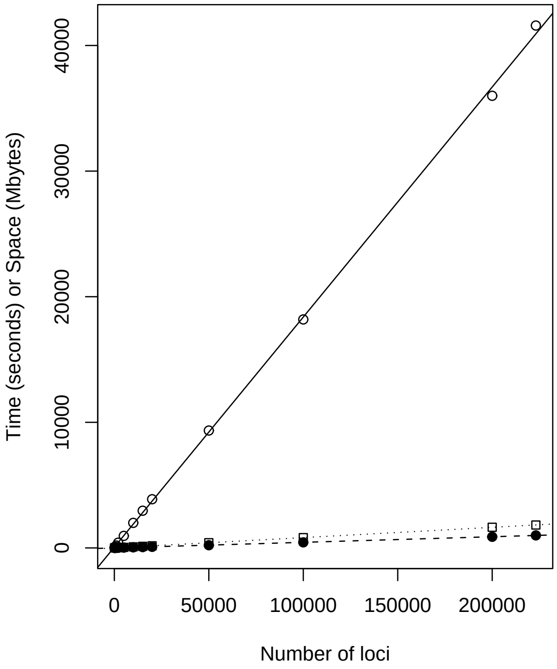


Fig. 4. The computational resources needed for LD model estimation and gene drop simulation plotted against the number of loci. The times needed for estimation are shown as white circles, the storage requirements are shown as white squares. The times needed for simulation are shown as black circles. The best linear fits are also plotted. The units for the vertical axis are seconds for the time data and Mbytes for the storage data.

loci available for chromosome 1. All the results described here were obtained using the author's lap top computer running Java 1.5.0_02 under Linux. The machine has two 2.8 GHz processors and 4 Gbytes of random access memory.

Figure 4 shows the times taken in seconds and the storage required in Mbytes for model estimation using `IntervalLD` with the default parameters described above. Also given are the times needed to make 100 gene drop simulations for a small three generation pedigree. The pedigree consisted of a sibship of 10 children, their parents and grandparents. The linear scaling for all these measures is clear.

Figure 4 shows how the resources required for LD model estimation change with the size of the sample. These results were obtained from artificial data sets obtained by duplicating and reduplicating the 60 individuals of the YRI data described above. In each case 10000 loci were modelled. This again shows clear linear scaling.

Thomas, 2009 showed that the haplotype frequencies implied by graphical models for LD were similar to those modelled by the `fastPHASE` program of Scheet and Stephens, 2006 . To further investigate this we tested the program's ability to impute missing genotypes from the haplotype model. For each of the first 600 loci, we deleted the genotype of one individual, so that 10 genotypes per individual were deleted in all. We then ran `IntervalLD` on this data and output the final genotypes imputed by the program for the missing values. The imputed values matched the actual values in 88.5% of cases. As a comparison, we also applied `fastPHASE` to the problem using its default parameter settings. This was correct 93.5% of the time.

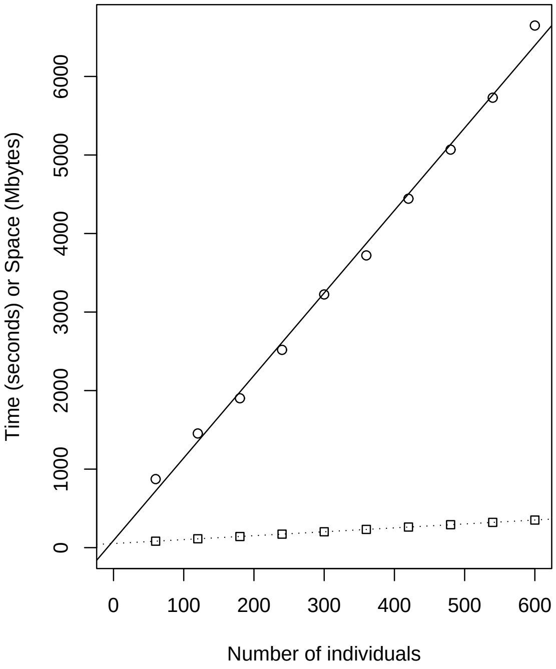


Fig. 5. The computational resources needed for LD model estimation for 10000 loci plotted against the number of individuals in the sample. As in figure 4, time is shown as white circles and storage as white squares. The best linear fits are again shown.

5 DISCUSSION

Current human SNP genotyping assays obtain genotypes on around one million loci for the individuals assayed. As the chromosomes segregate independently, modelling of LD and simulation can be performed separately for each chromosome. The largest number of loci that need to be considered jointly, therefore, are those on chromosome one, the largest chromosome. This number is under 100000 and well within the range considered here. The methods and programs described can therefore be applied to estimation and simulation problems on a genome wide scale. Model estimation of the data set of 223110 loci took just over 11 hours, from which we estimate that an analysis for a million loci would take around 50 hours. The limiting factor here is the storage space needed. Just over 1.8 Gbytes were required for the complete set of 223110 loci.

While the recent increases in the number of loci assayed is dramatic, the computational resources also, clearly, depend on the sample size. More generally we would expect the requirements to scale as $O(nmk)$ where

- n is the number of loci,
- m is the number of individuals assayed

- k is the average complexity of the graphical models considered.

The complexity of a graphical model on binary variables can be measured by $\sum_i 2^{c_i}$ where c_i is the number of variables in the ith clique, and the sum is over all cliques. To isolate the effects of increasing sample size and obtain figure 5, the α parameter of equation (1) was manipulated so that the model complexities remained similar. In reality, as the sample size increases weaker interactions may become detectable and the complexity of the graphical models may increase. Hence, in some circumstances super linear scaling may occur as the sample size grows. While this can be fixed by increasing α and enforcing parsimony, care should be taken not to oversimplify the models found. For larger samples, therefore, it might be necessary to break up the data further, perhaps handling the arms of the larger chromosomes separately.

Some experimentation with the parameters as used in the above comparison with the `fastPHASE` program, showed that results were not particularly sensitive to the choice of the the value for the maximum extent of the intervals. A maximum interval of 5 units each side of the locus allows the alleles at the locus to depend on up to 10 loci on each side. However, dependence on more than 5 was rarely seen. On the other hand the choice of α in equation (1) had a greater effect. The original setting was $\frac{1}{2}\log(h)$ where h is the number of chromosomes in the sample which is in accordance with the Bayesian information criterion of Schwarz, 1978 . However, better performance was seen with a far lower value that allowed for larger clique sizes. The default value is now set at $\alpha = \frac{1}{16}\log(h)$. We note that while the performance of `IntervalLD` on the imputation test gave good results, correctly imputing 88.5% of missing genotypes, this was not quite as good as `fastPHASE` which correctly imputed 93.5%.

Gene drop simulation has many possible applications, however, this work was primarily motivated by its use in assessing the statistical significance of allele sharing in relatives in identity by descent gene mapping strategies. Such methods have been described by Thomas *et al.*, 2008 and Leibon *et al.*, 2008 . These authors recognize that LD is likely to increase the lengths of random runs, or streaks, of allele sharing, potentially leading to false positive results. The methods and programs described here can be directly applied to this problem.

ACKNOWLEDGMENT

This work was supported by grants NIH R01 GM81417, and DOD W81XWH-07-01-0483 to Alun Thomas. I would like to thank three anonymous referees for their constructive criticisms and suggestions.

REFERENCES

Abecasis, G. R., Cherney, S. S., Cookson, W. O., and Cardon, L. R. (2001). Merlin - rapid analysis of dense genetic maps using sparse gene flow trees. *Nature Genetics*, **30**, 97–101.

de Berg, M., van Kreveld, M., Overmars, M., and Schwarzkopf, O. (2000). *Compuational Geometry. Algrorithms and Applications*. Springer-Verlag, second edition.

Dobra, A., Jones, B., Hans, C., Nevins, J., and West, M. (2003). Sparse graphical models for exploring gene expression data. *Journal of Multivariate Analysis*, **90**, 196–212.

Giudici, P. and Green, P. J. (1999). Decomposable graphical Gaussian model determination. *Biometrika*, **86**, 785–801.

Golumbic, M. C. (1980). *Algorithmic Graph Theory and Perfect Graphs*. Academic Press.

Greenspan, G. and Geiger, D. (2004). High density linkage disequilibrium mapping using models of haplotype block variation. *Bioinformatics*, **20, Suppl 1**, i137–i144.

Højsgaard, S. and Thiesson, B. (1995). BIFROST — Block recursive models Induced From Relevant knowledge, Observations, and Statistical Techniques. *Computational Statistics and Data Analysis*, **19**, 155–175.

Jones, B., Carvalho, C., Dobra, A., Hans, C., Carter, C., and West, M. (2005). Experiments in stochastic compuation for high-dimensional graphical models. *Statistical Science*, **20**, 388–400.

Kirkpatrick, S., Gellatt, Jr., C. D., and Vecchi, M. P. (1982). Optimization by simulated annealing. Technical Report RC 9353, IBM, Yorktown Heights.

Leibon, G., Rockmore, D. N., and Pollack, M. R. (2008). A snp streak model for the identification of genetic regions identical-by-descent. *Statistical Applications in Genetics and Molecular Biology*, **7**, 16.

MacCluer, J. W., Vandeburg, J. L., Read, B., and Ryder, O. A. (1986). Pedigree analysis by computer simulation. *Zoo Biology*, **5**, 147–160.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., and Teller, A. H. (1953). Equations of state calculations by fast computing machines. *Journal of Chemistry and Physics*, **21**, 1087–1091.

Scheet, P. and Stephens, M. (2006). A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase. *American Journal of Human Genetics*, **78**, 629–644.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, **6**, 461–464.

The International HapMap Consortium (2005). A haplotype map of the human genome. *Nature*, **437**, 1299–1320.

Thomas, A. (2005). Characterizing allelic associations from unphased diploid data by graphical modeling. *Genetic Epidemiology*, **29**, 23–35.

Thomas, A. (2007). Towards linkage analysis with markers in linkage disequilibrium. *Human Heredity*, **64**, 16–26.

Thomas, A. (2009). Estimation of graphical models whose conditional independence graphs are interval graphs and its application to modeling linkage disequilibrium. *Computational Statistics and Data Analysis*. Published on line.

Thomas, A. and Camp, N. J. (2004). Graphical modeling of the joint distribution of alleles at associated loci. *American Journal of Human Genetics*, **74**, 1088–1101.

Thomas, A., Camp, N. J., Farnham, J. M., Allen-Brady, K., and Cannon-Albright, L. A. (2008). Shared genomic segment analysis. Mapping disease predisposition genes in extended pedigrees using SNP genotype assays. *Annals of Human Genetics*, **72**, 279–287.

Verzilli, C. J., Stallard, N., and Whittaker, J. C. (2006). Bayesian graphical models for genomewide association studies. *American Journal of Human Genetics*, **79**, 100–112.

Assessment of SNP streak statistics using gene drop simulation with linkage disequilibrium

Alun Thomas[*]
Department of Biomedical Informatics
University of Utah

May 12, 2009

Running title: SNP streak statistics under LD.

Keywords: Genetic mapping, graphical modelling, identity by descent.

Abstract

We describe methods and programs for simulating the genotypes of individuals in a pedigree at large numbers of linked loci when the alleles of the founders are under linkage disequilibrium. Both simulation and estimation of linkage disequilibrium models are shown shown to be feasible on a genome wide scale. The methods are applied to evaluating the statistical significance of streaks of loci at which sets of related individuals share a common allele. The effects of properly allowing for linkage disequilibrium are shown to be important as they explain many of the large observations. This is illustrated by re analysis of a previously reported linkage of prostate cancer to chromosome 1p23.

[*] Genetic Epidemiology, 391 Chipeta Way Suite D, Salt Lake City, UT 84108, USA. alun@genepi.med.utah.edu, +1 801 587 9303 (voice), +1 801 581 6052 (fax).

1 Introduction

Dense single nucleotide polymorphism, or *SNP*, genotype assays offer both great opportunities and challenges to statistical genetics. The quantity, quality, and cheapness of the assays can allow powerful methods for precise gene localizations. However, the quantity of the data makes more involved forms of analysis intractable on a genome wide scale and the density of the loci assayed requires detailed modelling of the structure of the chromosomes. In particular, linkage disequilibrium, or *LD*, could for the most part be safely neglected when using micro satellite genotypes, whereas it it critical to model it accurately with dense SNP assays.

Although these challenges are encountered with population samples, they are even more problematic with pedigree data. Linkage analysis is a powerful statistical method that has been used very successfully to map disease genes, but it is computationally intense and intractable to carry out on one million marker genotype data. Moreover, adapting it to model LD greatly increases the computational requirements (Thomas 2007). This has led to in interest in simpler, more tractable analyses for family data that rely on the quantity of the data rather on statistical efficiency for their power. Thomas et al. (2008) introduced a mapping strategy based on counting long runs of loci at which individuals share alleles identical by state (*IBS*). Leibon et al. (2008) also considered the same statistic which they called the *SNP streak* statistic. Long runs of shared alleles should be rare in unrelated individuals, whereas in relatives within a pedigree, or in samples from founder populations, these long streaks indicate underlying sharing of genomic segments identical by descent (*IBD*) from a common ancestor. If the individuals sharing are selected to have some trait, and they are sufficiently distantly related that probability of random sharing is very low, then there is strong evidence that the shared segment contains a gene affecting the trait.

Houwen et al. (1994) and Heath et al. (2001) both studied relatively isolated founder populations to identify a small number of related cases who shared common chromosomal segments which they used to map disease genes. However, neither of these approaches incorporated precise pedigree relationships between the cases. Chapman & Thompson (2002) and te Meerman & Van der Meulen (1997) examined shared chromosomal segments in a founder population and showed how these are affected by the time since the founding of the population, population growth, genetic drift, selection and population subdivision. The streak statistic is similar to the haplotype sharing statistics of Van der Meulen & te Meerman (1997) and Beckmann et al. (2005), however, these more complicated statistics require genotyping of close relatives to estimate phase, are based on combining pairwise comparisons, and are applied in populations samples rather than in extended pedigrees. Bourgain et al. (2001) applied similar methods to extended pedigrees, but this again required knowing phase and combining pairwise comparisons. Other streak statistics have also been suggested, for example, Miyazawa et al. (2007) considered steaks of SNP loci at which individuals share homozygously. Other approaches exploiting IBD in pedigrees includes the homozygosity-by-descent method of Abney et al. (2002) which has been used to map recessive traits in known, very large, inbred populations (Newman et al. 2003).

Genomic sharing in a pedigree was modelled by Donnelly (1983) as a random walk on the vertices of a hypercube from which the distribution of number and length of genomic segments shared genome wide by an arbitrary set of relatives can be obtained. Cannings (2003) also derived results for this model. These results, however, describe the underlying process and do not account for observed genetic data. Both Thomas et al. (2008) and Leibon et al. (2008) addressed this problem using the streak statistic, although they differed in how they evaluated the statistical significance of observed streaks. Leibon et al. (2008) extended the theoretical results of Miyazawa et al. (2007) for homozygous sharing to the case where only one haplotype is shared, whereas, Thomas et al. (2008) used multi locus gene drop simulation. Both methods assumed that the loci being assessed were in linkage equilibrium, but both sets of authors also acknowledged that this assumption was inappropriate and likely to lead to underestimates of any p-value, and possible false positive results. It is this limitation that we seek to address here.

Thomas (2009*a*) and Thomas (2009*b*) developed and described methods and programs for modelling LD using restricted types of graphical models. These models are tractable so that both estimating them and simulating from them can be done on hundreds of thousands of loci. Given appropriate controls we can estimate LD models on a genome wide scale, and given a pedigree structure we can simulate founder haplotypes from the models and use the multi locus gene drop method to simulate genotypes for the entire pedigree. In what follows we apply these methods to a re analysis of the prostate cancer linkage to chromosome 1p23 reported by Camp et al. (2005) and previously analyzed using SNP streaks by Thomas et al. (2008). We show that appropriate modelling of LD is not only feasible on this scale, but that it is essential for accurately assessing statistical significance. While our focus here is on streak statistics observed in extended pedigrees, we note that the simulation method can also be applied to evaluate the significance the other statistics mentioned above, and to sampling designs using parent-offspring triplets, nuclear families or independent samples.

2 Materials and methods

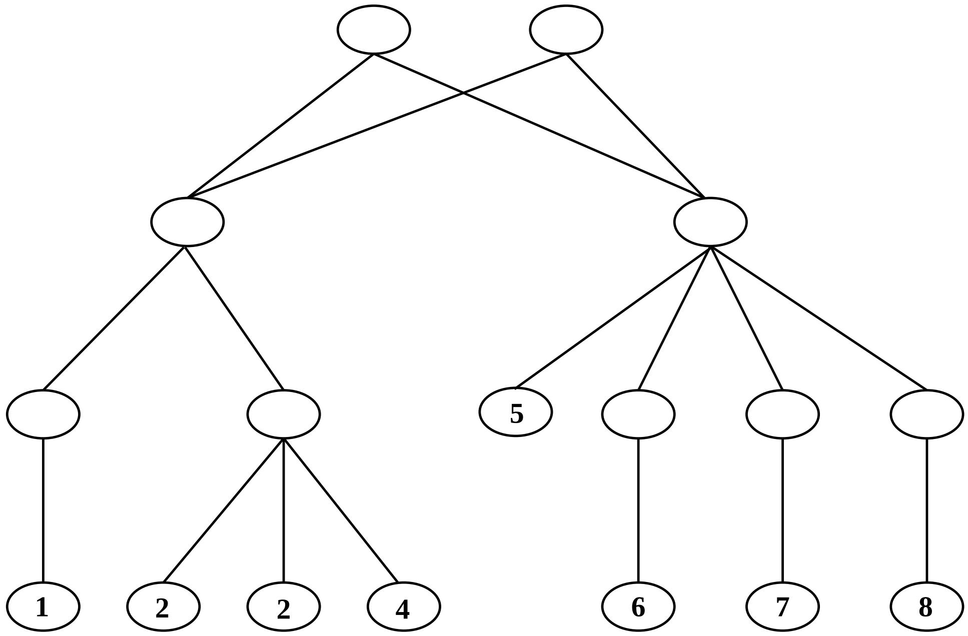
2.1 SNP streak statistics

Figure 1 shows a pedigree connecting 8 men with prostate cancer. These individuals were selected from a larger extended pedigree from a study of prostate cancer in Utah. These cases are connected by a total of 15 meioses to each of 2 recent common ancestors. Assuming that the total genetic length of the 22 autosomes is 35 Morgans (Broman et al. 1998), the probability that all 8 cases share any genetic segments IBD is 0.067 (Thomas et al. 2008).

Each of the 8 cases was genotyped by the Center for Inherited Disease Research, using the Illumina 110K panel (http://www.illumina.com). Of the total of 109299 loci analyzed, 9819 were on chromosome 1. In order to avoid spurious runs of sharing due to low heterozygosity, we discarded SNPs for which the heterozygosity score, as assessed from the controls described below, was less than 0.2. This left 8016 loci in the analysis, evenly spread over chromosome 1 with the exception of a large gap at the centromere.

At each locus i we counted the number of observations of each genotype: n^i_{11}, n^i_{12} and

Figure 1: A pedigree connecting 8 men with prostate cancer.



n^i_{22} so that $n^i_{11}+n^i_{12}+n^i_{22} \leq n = 8$, with inequality when there were some missing genotypes. Then we calculated the sharing statistic at the ith locus as

$$S_i = n - \min(n^i_{11}, n^i_{22}).$$

Then, again at each locus, we define $R_i(t)$ to be the longest run containing the locus for which the values of S_i are at least t. We took $t = n$ and $t = n - 1$ for this study, but sharing in smaller sets may appropriate if more cases are considered.

These statistics were calculated using a program written by the author which, like all the programs described here, is freely available from the author's web site, http://bioinformatics.med.utah.edu/∼alun. The program for calculating SNP streaks is named `SGS`, for shared genomic segments, and is called as follows:

```
java SGS input.par input.ped  > output
```

where

- `input.par` is a LINKAGE format parameter file describing the genetic loci.
- `input.ped` is a LINKAGE format pedigree file giving the pedigree structure and the genotypes of any assayed individuals. The individuals among who sharing is to be counted must have their proband status set to 1.
- `output` is a text file with one line for each marker in the input. Each gives the marker name followed by S_i, $R_i(n)$, $R_i(n-1)$, $R_i(n-2)$, and $R_i(n-3)$.

The maxima of the $R_i(t)$ statistics over the whole of the data being analyzed are also output to the screen.

Full details of the LINKAGE file formats can be had on the web at http://linkage.rockefeller.edu. This format is intended primarily for linkage and segregation analysis and so the distances between loci are specified as recombination fractions. Since it is useful also to have the physical location of the marker, we make the local convention of encoding this in the marker's name so that it is printed as part of the output from `SGS`. Note that all file names given here and below are just arbitrary examples, they are not required names.

2.2 LD model estimation

In order to estimate the LD structure of chromosome 1 we used the same control data that was described by Thomas et al. (2008). These are 52 Utah CEPH controls that were included as part of the 120 control sample set genotyped by Illumina for the same set of SNPs.

Graphical models (Lauritzen 1996) are a broad class of statistical models encompassing Bayesian networks, Markov random fields and probabilistic expert systems. They can tractably model complex relationships between variables. In particular, they can be efficiently estimated from data and used for simulation. The use of graphical models to model LD was first proposed by Thomas & Camp (2004) who developed a method of model estimation from phase known haplotype data. Thomas (2005) extended this to using unphased genotypic data by employing a two stage stochastic search approach. Given an initial model for LD, haplotypes are imputed conditional on the model and the genotype states. The imputed haplotypes are then used for re estimating the model using the Thomas & Camp (2004) method. The haplotypes are then re imputed, and so on. While this approach works well on moderately sized sets of loci, up to a few thousand, it does not scale well beyond this order of magnitude. Thomas (2009*a*), however, showed that restricting the class of models considered to that of interval graphs greatly improved efficiency without sacrificing modelling properties.

Thomas (2009*b*) describes an implementation of this approach called `IntervalLD` which employs a walking window approach so that the program's running time scales linearly with the number of loci being modelled. The program was run as follows:

```
java IntervalLD input.par control.ped > ldmodel
```

where

- `input.par` is the LINKAGE parameter file described above.
- `control.ped` is a LINKAGE pedigree file giving the genotypes of the control samples. The controls are unrelated so for each individual the parents are given as 0. If any relationships are specified in the control data, `IntervalLD` will ignore them and treat the observations as unrelated.

- `ldmodel` is a text file containing the estimated interval graphical model. The first line gives the number of alleles seen at each locus. This is followed by one line for each locus specifying the conditional distribution of alleles at the locus given the states at the loci that the program has estimated to be relevant. Although the file is human readable, it is primarily intended for input to other programs as described below.

2.3 Gene drop simulation with LD

Gene drop is a simple method for randomly generating genotypes for a set of related individuals. Alleles are randomly assigned to the founders of a pedigree and dropped at random to their offspring and other descendants mimicking Mendelian inheritance. The single locus method was described by MacCluer et al. (1986). The multi locus method is similar, but the inheritances at a genetic locus depend on those at the previous locus and the recombination fraction between them. An implementation of this is given, for instance, by the MERLIN program (Abecasis et al. 2001). The implementation given here differs only in that the alleles for the founders are simulated as haplotypes generated from the distribution specified by the given graphical model. The program `SimSGS` implements this to simulate random genotypes to match those in a given input file. This includes matching the pattern of missing data: if a genotype is missing in the real data it will also be missing in the simulated. The program is used as follows:

```
java SimSGS input.par input.ped s ldmodel > output
```

where

- `input.par` is the same LINKAGE parameter file as used above with `SGS` and `IntervalLD`.

- `input.ped` is the same LINKAGE pedigree file as was used to obtain the observed statistics using `SGS`. This will specify the pedigree and the genotypes to simulate.

- `s` is the number of simulations to perform. This parameter is optional, the default value is 1000.

- `ldmodel` is a graphical model for LD as estimated above using `IntervalLD`. This is also an optional parameter. If it is omitted the allele frequencies given in `input.par` are used to simulate data using conventional multi locus gene drop under the assumption of linkage equilibrium.

- `output` is a text file containing one line for each simulation made. On each line are the maxima over loci i of $R_i(n)$, $R_i(n-1)$, $R_i(n-2)$, and $R_i(n-3)$ for that simulation.

We also have a program `SimSGSRegions` which is used with the same syntax as `SimSGS` but the file `output` gives the number of times each locus is contained in the maximum run simulated.

3 Results

Figure 2 plots (a) $R_i(8)$ and (b) $R_i(7)$ for our case data on chromosome 1. The longest run for all 8 sharing was 64, and for 7 from 8 sharing was 495 occurring at the position marked A in the plots. This corresponds to the location previously found using the same method by Thomas et al. (2008) and to the original linkage peak reported by Camp et al. (2005) for the same family. Other locations with long runs are marked B, C, D and E on these and following plots. The distribution of locus heterozygosity is shown in figure 2(c) as a cumulative sum chart plotting $\sum_{j=1}^{i}(h_j - \bar{h})$ against the locus number i where $h_j = 2p_j(1 - p_j)$, $\bar{h} = \frac{1}{n}\sum_{j=1}^{n} h_j$, and p_j is the observed frequency of the minor allele at the jth locus.

We made three sets of simulations with which to evaluate the statistical significance of the observations. Two sets of 10000 simulations were made under the assumption of linkage equilibrium, the first assuming allele frequencies of 0.5 for each allele at each locus, the second using allele frequencies estimated from the controls. The first of these approaches is clearly something of a straw man and should not be used in reality, however, it gives us a base line with which to compare subsequent simulations. The third set of simulations was made under LD. The model fitting program `IntervalLD` uses a stochastic search, so the estimated model will typically differ from run to run. To see whether this had any influence on the results we made 10 independent LD model estimates from the control data and simulated 1000 gene drops from each of them. For each set of simulations we obtained the cumulative probability distribution of the maximum run length for IBS sharing between all 8, and between 7 of 8 cases. These results are shown in figure 3.

Finally, for three sets of simulations as described above we recorded the probability that each locus was contained in the longest run of sharing. The probabilities are plotted against location in figure 4. Note that as the maximum run lengths increase, more loci are covered by them, hence, the areas under the curves shown in figure 4 changes. The locations of the peaks seen in figure 2 are also indicated in these plots.

4 Discussion

Several of the issues raised in the simulation analysis of Thomas et al. (2008) have been addressed here. While the previous simulation program worked on the physical locations of the markers, `IntervalLD` works on a genetic map as specified by the recombination fractions given in the linkage parameter file. This allows specification of recombination hot and cold spots. The file format also allows different maps for male and female recombinations and the program interprets and uses these appropriately.

More importantly, we are now able to model LD in the simulations. The example presented here had 8016 loci. The average time taken for the 10 runs to estimate models on these loci was just under 400 seconds. However, the program has been demonstrated to scale linearly with the number of loci. Thomas (2009*b*) ran the program on over 200000 loci for 60 control individuals taking just under 700 minutes to complete the model estimation.

Figure 2: (a) shows $R_i(8)$, the run lengths where all 8 cases share an allele IBS plotted against the physical location of the loci. (b) shows $R_i(7)$, the run lengths where at least 7 of the 8 cases share an allele IBS. (c) is a cumulative sum chart for the heterozygosities of the loci: areas of steeply increasing or decreasing slope indicate high or low heterozygosity respectively.

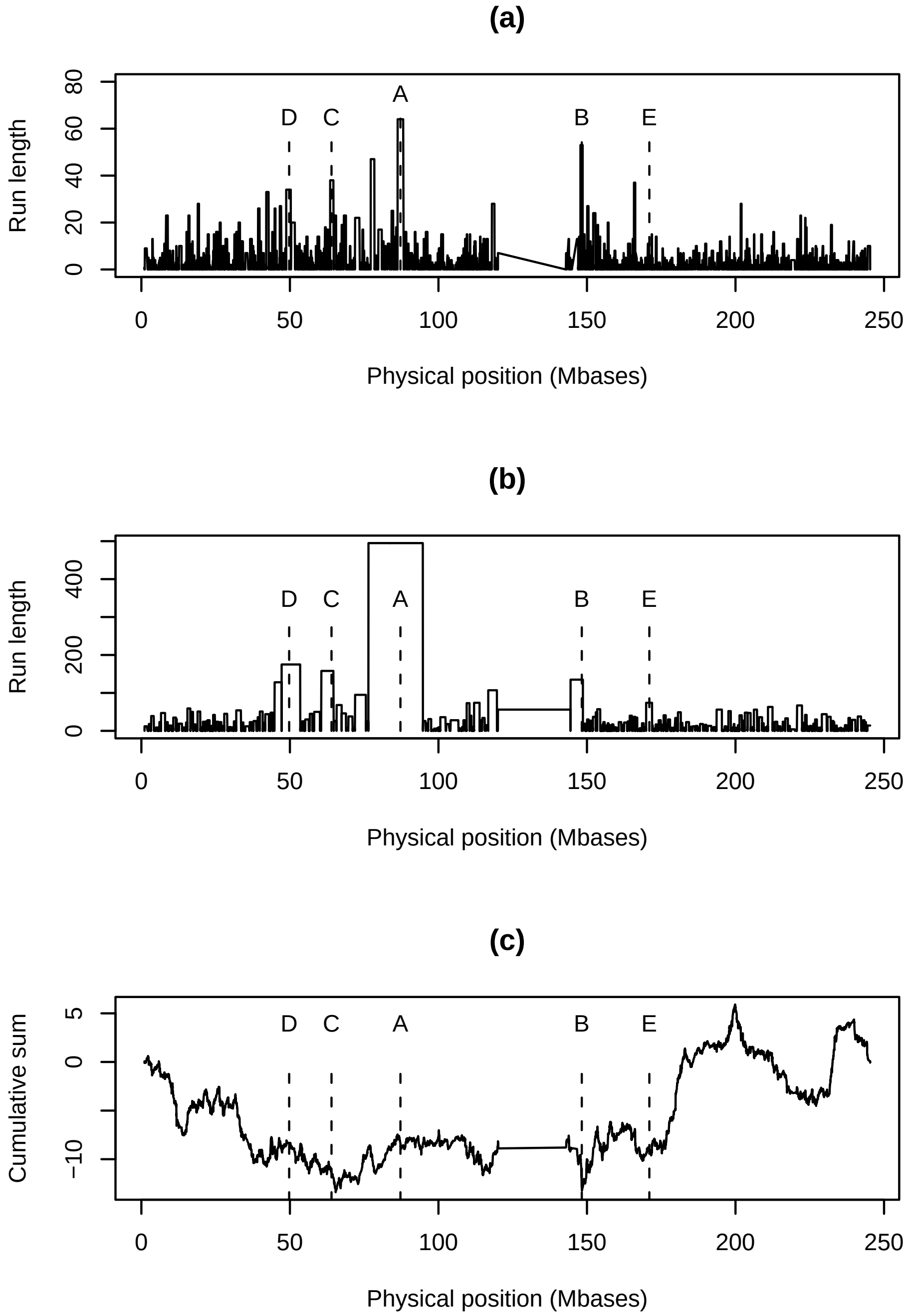


Figure 3: These plots show the cumulative distribution functions for the maximum simulated run lengths for which (a) all 8 cases share an allele IBS and (b) 7 of 8 cases share. In both plots the distribution furthest to the left is that from 10000 simulations when the loci are assumed to be in linkage equilibrium with equal allele frequencies. The distribution in the middle is that for 10000 simulations under linkage equilibrium, but with allele frequencies at each locus estimated from control data. On the right of each plot are overlaid 10 distributions each of 1000 simulations under a linkage disequilibrium model estimated from the control data. The dashed vertical lines show the values observed in the data.

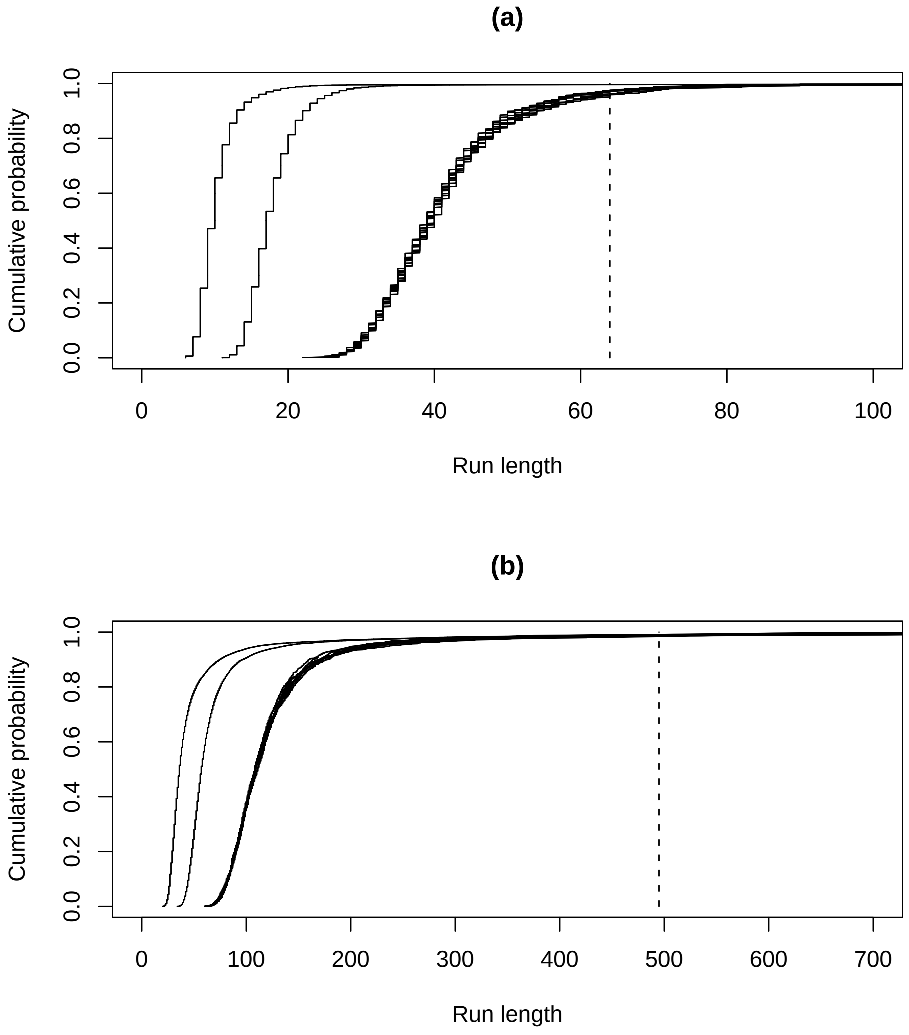
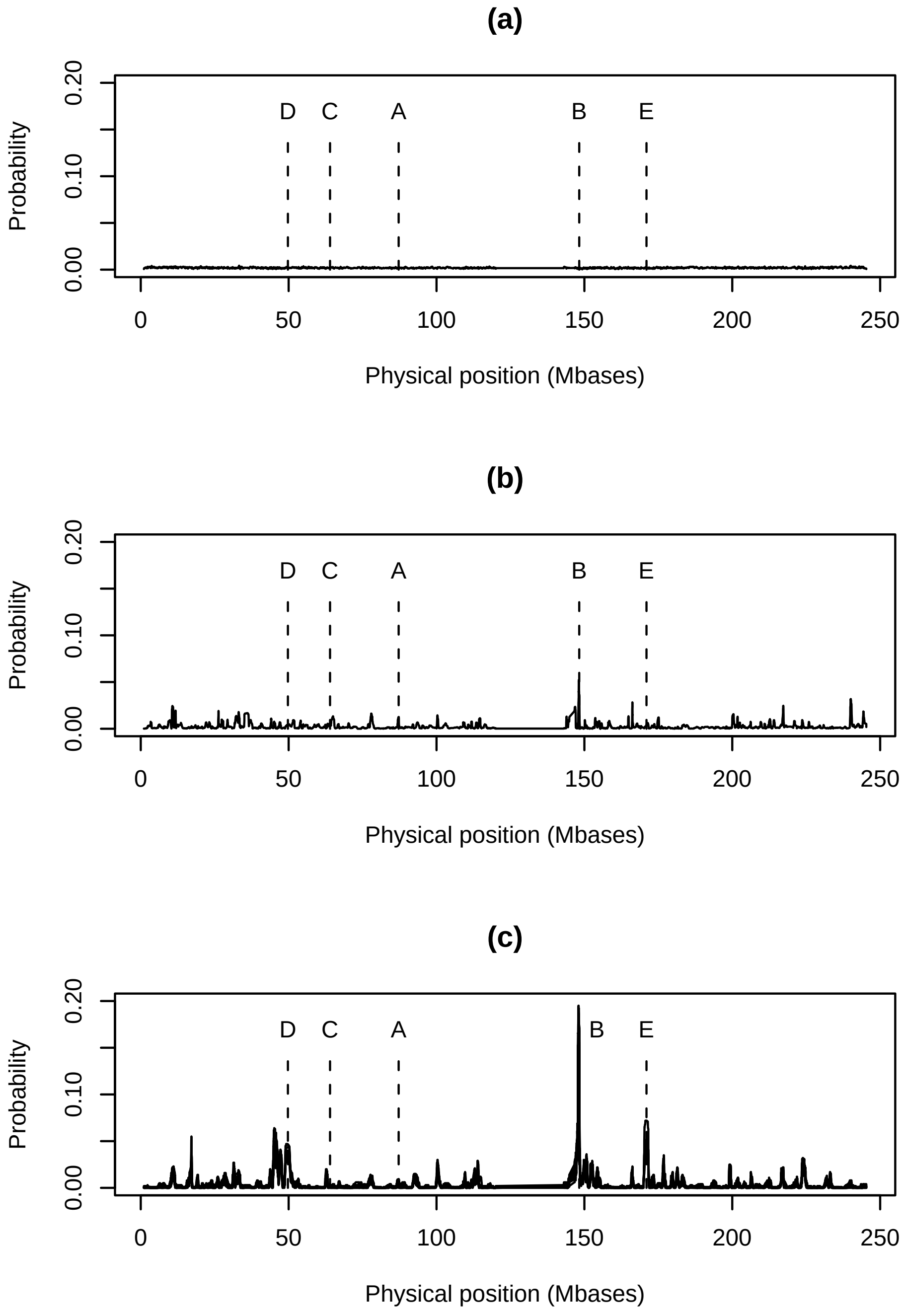


Figure 4: These plots show the probability, as estimated by simulation, that a locus is contained in the maximum run of sharing among all 8 cases. (a), (b) and (c) show respectively 10000 simulations under linkage equilibrium with equal allele frequencies, 10000 simulations under linkage equilibrium with estimated allele frequencies, and 10 x 1000 simulations under LD.



As the largest number of loci that need to be considered together in a genome wide analysis for 1 million loci are the approximately 100000 loci on chromosome 1, the program is well able to deal with this scale of analysis. Having estimated a graphical model for LD, the simulations themselves are reasonably quick. Making 1000 gene drops on the pedigree in figure 1 takes just under 300 seconds. Again, the gene drop program has been shown to scale linearly with the number of loci (Thomas 2009*b*). All the running times given here are for the author's HP laptop that runs Java 1.5.0_02-b09 under Linux. It has 4 Gbytes of memory and two 2.8 GHz central processing units.

Not only is large scale LD modelling feasible, it is also shown here to be necessary. The distributional shifts seen in figures 3 (a) and (b) due to LD are far greater than the shifts obtained by using realistic allele frequencies rather than assuming all alleles are equally frequent. The empirical p-value for a streak of 64 loci on chromosome 1 at which all individuals share an allele is 0.0037 under linkage equilibrium, but 0.034 under LD. The change for the run of 495 at which 7 of 8 share is less dramatic, from 0.01 to 0.012 which reflects the extreme nature of this observation: the length of sharing is beyond the influence of the LD. The empirical p-values under LD here are taken from combining the 10 samples of 1000 observations under the 10 different estimated LD models. However, the overlaid distributions in figure 3 show that although the models are typically different, the haplotypes estimated under them are very similar.

Figure 4 also demonstrates the dramatic effects of LD. Under linkage equilibrium with allele frequencies of 0.5, figure 4 (a) shows that the location of the longest simulated run length is uniformly distributed across the chromosome. When the control allele frequencies are used in the pedigree simulation, the longest run is more often at positions such as that marked B in the figures where there is a run of loci with low heterogeneity. The run of low heterogeneity at point B is demonstrated by the steep descent of the cumulative sum chart in figure 2 (c). Note also that since the longest runs are getting longer, more points are covered by them and there is, therefore, more area under the curve in figure 4 (b) than 4 (a). However, the peaks marked A, C, D and E in figure 2(a) do not correspond to points at which long runs are simulated under linkage equilibrium in figure 4 (b). The change from figure 4 (b) to (c) is even greater. We can now see that under LD, the peaks in the observed data at B, D and E all correspond to positions at which long runs of sharing are simulated. These peaks cannot, therefore, be taken as evidence of significant sharing due to selection for a common phenotype.

The conclusions for the validity of the linkage of prostate cancer to chromosome 1p23 are still mixed. The p-value for the run of 64 loci at which all 8 cases share on chromosome 1 is 0.034. This is not significant when we allow for selection of the peak on chromosome 1 as the best of all those seen in a genome wide analysis. Neither is the run of 495 loci at which 7 from 8 share. A more detailed inspection of the data in this region shows that it is the same individual who does not share each side of the 64 shared by all. Although there is clearly an underlying genomic segment shared here IBD by these individuals, this is not sufficiently unusual in the pedigree to indicate that the sharing is due to selection for phenotype. On the other hand, we note that the longest runs are rarely simulated at position A. This is unlike positions B, D and E which are clearly peaks due to low heterozygosity and LD, and unlike position C where the run length is near the median of the distribution of the maximum.

Since this is a re analysis of the pedigree that gave the original lod score of 3.1 reported by Camp et al. (2005), it can not serve as an independent assessment of the finding. It does, however, give an indication that the the pattern of IBD sharing in the pedigree is consistent with linkage. For true confirmation for the result more data is needed.

The pedigree design used here is suited to detecting genes with dominant mode of expression. For recessive diseases inbred pedigrees could be used, as could random samples from an inbred population. In this case we would define streaks as runs of loci where individuals share both alleles in common, as described by Miyazawa et al. (2007). It is also informative to look for streaks of loci where sampled individuals are homozygous, but not necessarily for the same haplotype, as this may actually indicate hemizygosity and the presence of a chromosomal deletion. In either case, the statistical significance can be assessed by simulations that need to take LD into account. Thus, the estimation and simulation programs described here are directly applicable.

While this work goes some way to addressing the effects of LD on streak statistics, there are other issues. Perhaps the most pressing is the sensitivity of streak statistics to genotyping error. A single misclassification can, potentially, end a streak. The presence of two longer than average runs adjacent to each other, as occasionally seen in our analyses, would suggest that underlying these is an even longer run that has been broken up by genotyping error. If we take the S_i as our basic statistics, then the problem is essentially one of detecting the change points in their distribution: the more individuals that share a genomic segment, the higher S_i should be on average, even in the presence of genotyping error. One approach to this may be to replace the R_i statistics with more robust methods from the field of process control. For instance, cumulative sum methods, or CUSUM charts, could be used with significance again being assessed using simulation under LD. This is an approach that we wish to investigate in future work.

5 Acknowledgments

This work was supported by Grant Number R01 GM081417 to Alun Thomas from the NIH National Institutes of General Medical Sciences, and NIH National Cancer Institute grant R01 CA90752 and subcontract from Johns Hopkins University with funds provided by grant R01 CA89600 from the NIH National Cancer Institute to L A Cannon Albright. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of General Medical Sciences, National Cancer Institute or the National Institutes of Health. It was also supported by US Army Medical Research and Material Command W81XWH-07-1-0483 to Alun Thomas.

Research was supported by the Utah Cancer Registry, which is funded by contract N01-PC-35141 from the National Cancer Institute's SEER program with additional support from the Utah State Department of Health and the University of Utah

Partial support for all datasets within the Utah Population Database was provided by the University of Utah Huntsman Cancer Institute.

This investigation was supported by the Public Health Services grant number M01-RR00064 from the National Center for Research Resources.

Genotyping services were provided by the Center for Inherited Disease Research which is fully funded through a federal contract from the NIH to the Johns Hopkins University, contract number N01-HG-65403.

References

Abecasis, G. R., Cherney, S. S., Cookson, W. O. & Cardon, L. R. (2001), Merlin - rapid analysis of dense genetic maps using sparse gene flow trees, *Nature Genetics* **30**, 97–101.

Abney, M., Ober, C. & McPeek, M. S. (2002), Quantitative-trait homozygosity and association mapping and empirical genomewide significance in large, complex pedigrees: Fasting serum-insulin levels in the Hutterites, *American Journal of Human Genetics* **70**, 920–934.

Beckmann, L., Thomas, D. C., Fischer, C. & Chang-Claude, J. (2005), Haplotype sharing analysis using Mantel statistics, *Human Heredity* **59**, 67–78.

Bourgain, C., Genin, E., Holopainen, P., Musthlahti, K., Maki, M. & Partanen, J. (2001), Use of closely related affected individuals for the genetic study of complex diseases in founder populations, *American Journal of Human Genetics* **68**, 154–159.

Broman, K. W., Murray, J. C., Sheffield, V. C., White, R. L. & Weber, J. L. (1998), Comprehensive human genetic maps: inidividual an sex-specific variation in recombination, *American Journal of Human Genetics* **63**, 861–869.

Camp, N. J., Swensen, J., Horne, B. D., Farnham, J. M., Thomas, A., Cannon-Albright, L. A. & Tavtigian, S. V. (2005), Characterizaion of linkage disequilibrium structure, mutation history, and tagging SNPs, and their use in association analyses: ELAC2 and familial early-onset prostate cancer, *Genetic Epidemiology* **28**, 232–243.

Cannings, C. (2003), The identity by descent process along the chromosome, *Human Heredity* **56**, 126–130.

Chapman, N. H. & Thompson, E. A. (2002), The effect of population history on the lengths of ancestral chromosome segments, *Genetics* **162**, 449–458.

Donnelly, K. P. (1983), The probability that related individuals share some section of the genome identical by descent, *Theoretical Population Biology* **23**, 34–63.

Heath, S., Robledo, R., Beggs, W., Feola, G., Parodo, C., Rinaldi, A., Contu, L., Dana, D., Stambolian, D. & Siniscalco, M. (2001), A novel approach to search for identity by descent in small samples of patients and controls from the same Mendelian breeding unit: a pilot study in myopia, *Human Heredity* **52**, 183–190.

Houwen, R. H. J., Baharloo, S., Blankenship, K., Raeymaekers, P., Juyn, J., Sandkuijl, L. A. & Freimer, N. B. (1994), Genomic screening by searching for shared segments: mapping a gene for benign recurrent intrahepatic cholestasis, *Nature Genetics* **8**, 380–386.

Lauritzen, S. L. (1996), *Graphical Models*, Clarendon Press.

Leibon, G., Rockmore, D. N. & Pollack, M. R. (2008), A SNP streak model for the identification of genetic regions identical-by-descent, *Statistical Applications in Genetics and Molecular Biology* **7**, 16.

MacCluer, J. W., Vandeburg, J. L., Read, B. & Ryder, O. A. (1986), Pedigree analysis by computer simulation, *Zoo Biology* **5**, 147–160.

Miyazawa, H., Kato, M., Awata, T., Khoda, M., Iwasa, H., Koyama, N., Tanaka, T., Huqun, Kyo, S., Okazaki, Y. & Hagiwara, K. (2007), Homozygosity haplotype allows a genomewide search for the autosomal segments shared among patients, *American Journal of Human Genetics* **80**, 1090–1102.

Newman, D. L., Abney, M., Dytch, H., Parry, R., McPeek, M. S. & Ober, C. (2003), Major loci influencing serum triglyceride levels on 2q14 and 9p21 localized by homozygosity-by-descent mapping in a large Hutterite pedigree, *Human Molecular Genetics* **12**, 127–144.

te Meerman, G. J. & Van der Meulen, M. A. (1997), Genomic sharing surrounding alleles identical by descent: Effects of genetic drift and population growth, *Genetic Epidemiology* **14**, 1125–1130.

Thomas, A. (2005), Characterizing allelic associations from unphased diploid data by graphical modeling, *Genetic Epidemiology* **29**, 23–35.

Thomas, A. (2007), Towards linkage analysis with markers in linkage disequilibrium, *Human Heredity* **64**, 16–26.

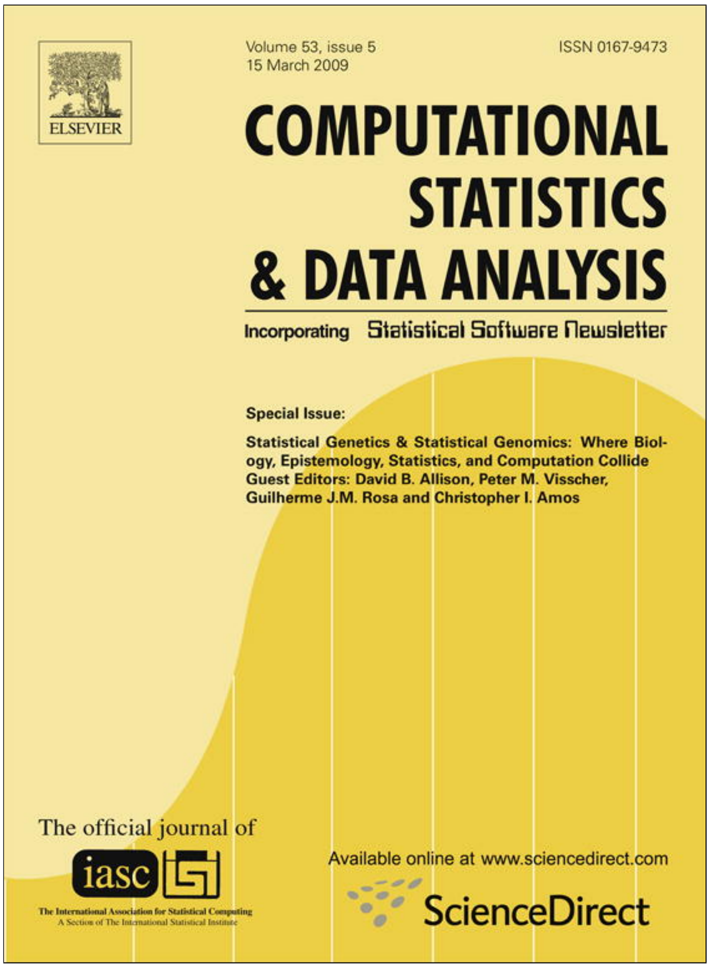
Thomas, A. (2009*a*), Estimation of graphical models whose conditional independence graphs are interval graphs and its application to modeling linkage disequilibrium, *Computational Statistics and Data Analysis* **53**, 1818–1828.

Thomas, A. (2009*b*), A method and program for estimating graphical models for linkage disequilibrium that scale linearly with the number of loci, and their application to gene drop simulation, *Bioinformatics*. In press.

Thomas, A. & Camp, N. J. (2004), Graphical modeling of the joint distribution of alleles at associated loci, *American Journal of Human Genetics* **74**, 1088–1101.

Thomas, A., Camp, N. J., Farnham, J. M., Allen-Brady, K. & Cannon-Albright, L. A. (2008), Shared genomic segment analysis. Mapping disease predisposition genes in extended pedigrees using SNP genotype assays, *Annals of Human Genetics* **72**, 279–287.

Van der Meulen, M. A. & te Meerman, G. J. (1997), Haplotype sharing analysis in affected individuals from nuclear families with at least one affected offspring, *Genetic Epidemiology* **14**, 915–919.



Volume 53, issue 5
15 March 2009

ISSN 0167-9473

ELSEVIER

COMPUTATIONAL STATISTICS & DATA ANALYSIS

Incorporating Statistical Software Newsletter

Special Issue:

Statistical Genetics & Statistical Genomics: Where Biology, Epistemology, Statistics, and Computation Collide
Guest Editors: David B. Allison, Peter M. Visscher, Guilherme J.M. Rosa and Christopher I. Amos

The official journal of
iasc
The International Association for Statistical Computing
A Section of The International Statistical Institute

Available online at www.sciencedirect.com
ScienceDirect

Available online at www.sciencedirect.com

ScienceDirect

Computational Statistics and Data Analysis 53 (2009) 1818–1828

COMPUTATIONAL STATISTICS & DATA ANALYSIS

www.elsevier.com/locate/csda

Estimation of graphical models whose conditional independence graphs are interval graphs and its application to modelling linkage disequilibrium

Alun Thomas*

Genetic Epidemiology, University of Utah, 391 Chipeta Way Suite D, Salt Lake City, UT 84108, USA

Available online 10 February 2008

Abstract

Estimation of graphical models whose conditional independence graph comes from the general class of decomposable graphs is compared with estimation under the more restrictive assumption that the graphs are interval graphs. This restriction is shown to improve the mixing of the Markov chain Monte Carlo search to find an optimal model with little effect on the haplotype frequencies implied by the estimates. A further restriction requiring intervals to cover specified points is also considered and shown to be appropriate for modelling associations between alleles at genetic loci. As well as usefully describing the patterns of associations, these estimates can also be used to model population haplotype frequencies in statistical gene mapping methods such as linkage analysis and association studies.

1. Introduction

It is important in many type of statistical analyses in genetics to have accurate models for the frequency distributions of alleles or haplotypes at genetic loci of individuals in a population. This is true both for association testing involving unrelated cases and controls, and for linkage analysis where the founders of a pedigree are assumed to be randomly selected from the population at large. In each of these cases, using inappropriate distributions can lead to both the loss of power to detect the effects of genes on phenotypes, and to excessive false positive gene findings (Amos et al., 2006). For an analysis using a single genetic locus, the requirement is simply a good estimate of the allele frequencies at that locus in the appropriate population. If a set of loci are used, and they are spaced sufficiently far apart, it is reasonable to assume that they are in *linkage equilibrium* in which case the probability of a haplotype is the product of the individual locus allele probabilities. However, if loci are densely spaced along the genome, strong correlations will exist between alleles at nearby loci. That is, the loci are in *linkage disequilibrium (LD)* (Ott, 1985). Failure to incorporate LD in the modeled joint distribution of alleles will distort the frequencies of haplotypes and can again result in excessive false positives and loss of statistical power in linkage or association analyses. Other types of analysis, such as mapping

* Tel.: +1 801 587 9303; fax: +1 801 581 6052.
E-mail address: alun@genepi.med.utah.edu.

0167-9473/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2008.02.003

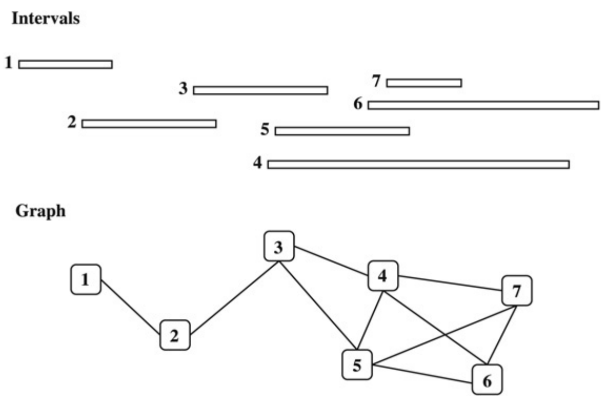


Fig. 1. An interval set and its corresponding interval graph.

based on segments of DNA shared between distant relatives (Thomas et al., 2008) also need to model haplotype frequencies accurately.

Methods developed to estimate general multi locus LD models that can be used in genetic mapping include that of Stephens et al. (2001), as implemented in the PHASE program and developed in the more recent FASTPHASE program (Scheet and Stephens, 2006), and general graphical modelling implemented in the HapGraph program (Thomas and Camp, 2004; Thomas, 2005). Both these approaches, while computationally intensive, can be applied to hundreds of loci, and can be incorporated into linkage and association analyses (Thomas, 2005, 2007). However, current genotyping assays involve over a million loci genome wide which makes the modeling problem intractable with current methods, even when the data is broken down to smaller components such as chromosome arms. In this paper, we consider the effects on computational tractability and model accuracy of modifications to graphical modelling methods which limit the class of graphs possible.

When the joint distribution of a set of random variables implies many independences or conditional independences between subsets of the variables, it can often be usefully considered as a graphical model. A graphical model has two elements: a *conditional independence*, or *Markov* graph, G, that represents the structure of the relationships between the variables and a set of parameters, M. If the distribution of $X_1, \ldots X_n$ factorizes as

$$P(X_1, \ldots X_n) = \prod_i f(T_i) \quad \text{where } T_i \subset \{X_1, \ldots X_n\}, \tag{1}$$

the vertices of the Markov graph are the variables $X_1, \ldots X_n$ with edges connecting pairs of variables if they appear together in one or more of the T_i. While the structure of a graphical model is often apparent from modelling assumptions, it is also possible to estimate it from a set of multivariate observations. This was originally developed by Højsgaard and Thiesson (1995) with more recent work by Giudici and Green (1999) and Thomas and Camp (2004) on continuous and discrete variables respectively. In all this work models are restricted to the class of *decomposable* graphical models that are well behaved, tractable and flexible. This class is defined and the main features of estimation methods are described below. The Markov graph of a decomposable model is a *decomposable graph*. Both Giudici and Green (1999) and Thomas and Camp (2004) use stochastic search methods for finding an optimal model, or Markov chain Monte Carlo (*MCMC*) methods for sampling from the posterior distribution of models. In each of these cases it is necessary, given a decomposable graph G to propose a new graph G' and accept or reject it as the new incumbent according to appropriate probabilities. If G' is decomposable, then Giudici and Green (1999) have shown that the value of the target function for the proposed model can be found very quickly, in time independent of the size of the graph. However, it is not straightforward to ensure the decomposability of G' in advance so that it is necessary to check for this condition and reject graphs that are not decomposable. As we show below, this rejection step makes this general approach extremely inefficient when the number of variables is large. However, it can be avoided if we restrict the graphs considered to those in a more manageable subclass: the class of *interval graphs*.

A graph is an interval graph if its vertices can be made to correspond to subintervals of the real line with pairs of vertices joined by an edge if their corresponding intervals overlap. This is illustrated in Fig. 1 and described more fully below. As Golumbic (1980) shows, all interval graphs are decomposable. If we now work with a set of intervals, one for each of the random variables in our model, it is easy to perturb these by moving and resizing them and yet be sure to stay within the class of interval graphs. If, furthermore, we find that the restriction to the set of interval graphs does not seriously affect our ability to accurately model the data, then we have a simpler and more computationally efficient estimation method.

Although this idea is developed generally to model associations between variables in any context, heuristically, it seems particularly appropriate for LD modelling. On average, according to Malecot's model, pairwise LD decreases as the distance between loci increases (Morton, 2002), however, on a fine scale, more complex patterns appear. Thomas (2007) showed that, at least when dealing with small genomic regions, the haplotype frequencies estimated by both the PHASE program and graphical modelling are quite different to those obtained by fitting low order Markov models. Because of the linear arrangement of genetic loci along a chromosome, and the expectation that LD decreases with distance, modeling with interval graphs has clear intuitive appeal: most statistical geneticists have some informal notion of the *extent* of LD around a locus. In what follows, therefore, we consider not only the complete class of interval graphs which may have general applications, but also a more constrained subclass which will be appropriate when there is some reason to expect that a linear arrangement of the variables affects correlation.

2. Methods

2.1. Estimating graphical models

Consider a graph $G = G(V, E)$ with vertices V and edges E. A subset of vertices $U \subseteq V$ defines an *induced subgraph* of G which contains all the vertices U and any edges in E that connect vertices in U. A subgraph induced by $U \subseteq V$ is *complete* if all pairs of vertices in U are connected in G. A *clique* is a complete subgraph that is maximal, that is, it is not a subgraph of any other complete subgraph.

A graph G is *decomposable* if and only if the set of cliques of G can be ordered as $(C_1, C_2, \dots, C_c)$ so that

$$\text{if } S_i = C_i \cap \bigcup_{j=i+1}^{c} C_j \text{ then } S_i \subset C_k \quad \text{for some } k > i. \tag{2}$$

This is called the *running intersection property*. This condition is equivalent to requiring that the graph is *triangulated*, or *chorded*, (Golumbic, 1980), that is, it contains no unchorded cycles of four or more vertices. The sets S_i are called the *separators* of the graph, and although several orderings typically give the running intersection property the cliques and separators are uniquely determined by the graph structure.

A graphical model with a decomposable Markov graph is a *decomposable model* and joint distribution of the variables in the model can be decomposed in terms of the marginal distributions of the cliques and separators:

$$P(X_1, \dots X_n) = \prod_i \frac{P(C_i)}{P(S_i)}. \tag{3}$$

For discrete variables these marginals are simple multinomials, and so, given a set of observations, it is straightforward to calculate maximum likelihood estimators of the parameters, the maximized likelihood and the degrees of freedom. Multivariate Gaussians are similarly tractable in the continuous case. The decomposability then allows us to combine these to obtain the overall maximized log likelihood and degrees of freedom

$$\log \hat{L}(G) = \sum_i \log \hat{L}(C_i) - \sum_i \log \hat{L}(S_i), \tag{4}$$

and

$$\mathrm{df}(G) = \sum_i \mathrm{df}(C_i) - \sum_i \mathrm{df}(S_i). \tag{5}$$

Model estimation can then be based on optimizing a penalized likelihood *information criterion*

$$IC(G) = \log \hat{L}(G) - \alpha \, \mathrm{df}(G), \tag{6}$$

where α is some arbitrary constant. Højsgaard and Thiesson (1995) use a deterministic optimization while Giudici and Green (1999) and Thomas and Camp (2004) use stochastic search or sampling methods. The stochastic methods require that an incumbent decomposable graph G is perturbed, for example, by adding or deleting an edge, to give a proposed new graph G'. If G' is not decomposable it is immediately discarded, otherwise it is accepted or rejected with the appropriate probabilities for Metropolis (Metropolis et al., 1953) or Hastings (1970) sampling, or simulated annealing optimization (Kirkpatrick et al., 1982). Giudici and Green (1999) give very fast methods for evaluating the rejection probability whose computational requirements do not increase with the number of variables being considered. Their algorithm for determining whether G' is decomposable can take order n time in the worst case, but in practice is very quick. However, for large graphs, the probability that a random perturbation to G will result in decomposable G' is small. For instance, if we consider adding or subtracting an edge there are $n(n-1)/2$ pairs of vertices to choose from, whereas, intuitively, we would expect $O(n)$ of these flips to result in a decomposable proposal.

2.2. *Interval graphs*

A graph is an interval graph if its vertices can be made to correspond to intervals of the real line and its edges connect pairs of vertices if and only if the corresponding intervals overlap. This is illustrated in Fig. 1. Intuitively, an interval graph would be expected to be long and thin, and this is indeed the case: this notion can be formalized in terms of the longest path in the graph and how far a vertex can be from this path (Golumbic, 1980). Moreover, an interval graph is always decomposable. Thus, if we restrict our search for decomposable models to those with interval Markov graphs, we can work with the more tractable interval representations of the graphs instead of the graphs themselves. Whatever perturbations to the solution then involve, for example, moving an interval or changing its length or more complex manipulations involving multiple intervals, the result will always give an interval graph and a decomposable model. The benefits of this can be twofold. First, the perturbations can be more radical than simply adding or deleting an edge and so can potentially give better mixing properties for the sampler or optimizer. Second, we do not need to waste time proposing nondecomposable solutions.

It should be recognized, however, that we are sampling interval sets, not graphs directly. Since interval graphs can be represented as interval sets in different numbers of ways, those graphs with more interval set representations will be oversampled, and those with fewer will be undersampled. While this might be accounted for in the Metropolis or Hastings rejection probability, we will assume that this effect is small when we are sampling graphs of similar probability, and justify this empirically below.

2.3. *Efficient implementation*

In order to take advantage of this idea, we need two things. One is to have a data structure that allows interval sets to be managed and queried efficiently. The other is to be able to evaluate the maximized log likelihood and degrees of freedom of a proposed model quickly, and preferably in time that does not depend on the size of the problem.

The first issue is resolved by using a standard data structure called an *interval tree* (de Berg et al., 2000). The root of the tree is associated with a fixed point, typically the midpoint of a finite region that contains all the intervals. This root node stores a list of the intervals that cover the fixed point. All intervals that lie completely to the left of the point are delegated to a daughter node whose fixed point is the mid point of the left region, and similarly for intervals which lie completely to the right of the fixed point. The structure is built up recursively in this way until all intervals are stored in a list at one of the nodes in the tree. This structure allows addition of new intervals, deletion of existing intervals, querying for intervals that cover a particular point, and querying for intervals that overlap with a given interval to be done in $O(\log n)$ time.

To address the second issue of efficient likelihood recalculation, we first note that the set of intervals that cover any point on the line correspond to a *complete cutset* of the graph (Golumbic, 1980). A set of vertices K is a cutset if it partitions the vertices of G into L, M and K itself such that all paths in G from a vertex in L to a vertex in M must pass through a vertex in K. The separators S_i of G are all complete cutsets, in fact, all the minimal complete cutsets. The complete cutsets defined by points on the line will include these separators and also complete cutsets that are not minimal.

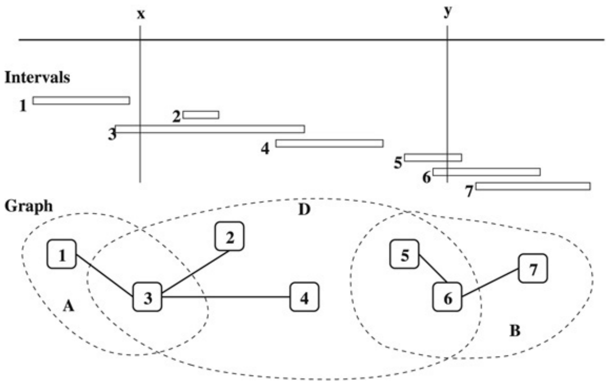


Fig. 2. A subregion partitions the interval graph.

For any graphical model, if K is a complete cutset then the variables L are conditionally independent of M given the value of K. That is

$$P(KLM) = P(L|K)P(M|K)P(K) = \frac{P(LK)P(MK)}{P(K)}. \tag{7}$$

If we now consider a subregion (x, y) of the line we can define three induced subgraphs of G: A, B and D the subgraphs induced by the intervals that overlap with $(-\infty, x)$, (y, ∞) and (x, y) respectively, so that $A \cap D$ and $B \cap D$ will be the complete cutsets defined by the intervals that cover the points x and y. This is illustrated in Fig. 2. The subregion (x, y) thus defines conditional independences that can be expressed as

$$P(ABD) = \frac{P(A)P(B)P(D)}{P(A \cap D)P(B \cap D)}. \tag{8}$$

If we now alter the graph G to make G' in such a way that only intervals that lie completely in (x, y) are changed, D may change to D' but A and B will not be affected. Moreover, $A \cap D' = A \cap D$ and $B \cap D' = B \cap D$. Hence,

$$\frac{P(G')}{P(G)} = \frac{P(A)P(B)P(D')}{P(A \cap D')P(B \cap D')} \times \frac{P(A \cap D)P(B \cap D)}{P(A)P(B)P(D)} = \frac{P(D')}{P(D)}. \tag{9}$$

In this way, the change in the global joint probability can be evaluated very quickly from local changes.

As with Eq. (3), this extends to allow us to quickly evaluate changes in the maximized log likelihood and degrees of freedom, and hence the information criterion $IC(G')$. So, for perturbations of G that involve changing just one interval, we need only consider the graph corresponding to the portions of the line that lie under the interval before it is changed and after it is changed. Hence, we can very efficiently evaluate the target function for the proposed graph G'.

In our implementation of this scheme, intervals are initially allocated with midpoints evenly distributed between 0 and 1, with small lengths so that no intervals overlap. Perturbations consist of randomly choosing an interval and either giving it a new midpoint chosen uniformly at random in (0, 1), or giving it a new length chosen from an exponential distribution, or both.

2.4. Constrained interval graphs

When the variables being modelled can be ordered linearly it may be appropriate to reflect this in the structure of the interval graph. For example, genetic loci have physical positions along a chromosome and we strongly expect the greatest correlations to be between alleles at loci that are nearest each other. In this case we can require the interval that represents a particular locus to cover its physical location. We also alter the definition of the graph to require intervals to overlap by some minimal amount in order to add an edge between the corresponding vertices. Any vertex corresponding to an interval of length less than this minimal amount will therefore not be connected to any other vertices. This is illustrated in Fig. 3. This extra condition gives some flexibility to the model. For example, with reference to Fig. 3, suppose that locus 2 appears from the data to be independent of all other loci, but that loci 1 and

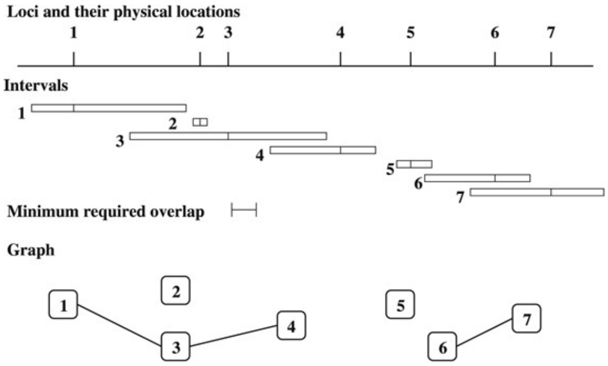


Fig. 3. An interval graph constrained by the physical location of genetic loci.

3, and 3 and 4 are very strongly correlated. Without this final requirement, the interval structure would force an edge between 2 and 3 making the model more complex than necessary. Such a situation may often arise with genetic loci where the frequency of the less frequent allele is very low. It is trivial to show that requiring a minimal overlap still gives an interval graph.

In this case the intervals are initially set as for the general interval graph. Perturbations involve randomly extending or reducing the spans to each side of the required fixed point by amounts generated from and exponential distribution.

This approach can also be used if an ordering of the variables is know but distances are not. In this case we can assign the variables to arbitrary evenly spaced points along the line.

2.5. Programs

General and constrained interval graph searches have been incorporated into the author's HapGraph program (Thomas and Camp, 2004) which can be used both as a generic graphical model estimator, or for the specific case of modelling allelic association. This latter case requires an extra step to account for observing unordered genotypes as opposed to complete phase known haplotypes. Both versions allow for missing data by random imputation. Full details of the methods are given by Thomas (2005). The program is written completely in Java thus is platform independent, and can be obtained from http://bioinformatics.med.utah.edu/~alun.

3. Results

We illustrate the effects of the model restrictions described here using data for subsets of the single nucleotide polymorphisms on chromosome 1 genotyped in the sample of 60 Yoruba people from Ibadan, Nigeria by the HapMap project (The International HapMap Consortium, 2005). This sample is conventionally abbreviated as YRI, and the data was from build 36 dated 2 May 2007. The loci that were monomorphic in this sample were not considered in these analyses. We used subsets of the first 20,000 remaining loci in what follows.

In order to first consider the computational effects of model restrictions we ran three versions of the HapGraph program. The first fitted a general decomposable graph using the rejection method of Giudici and Green (1999), which is the standard form of the program. The other two implemented a general interval graph and a constrained interval graph search as described above. HapGraph's graphical user interface that shows the graph as it is being updated was not used so as to avoid incorporating the processor time needed for graphical rendering in the comparisons. Fig. 4 shows the times taken by each of the three methods to perform one million Metropolis updates of the graph for data on sets of between 20 and 20,000 loci. Fig. 5 plots the largest penalized log likelihood score seen in each of the runs. All the programs were run on the author's laptop computer which has a 2.33 GHz dual core central processing unit running Red Hat Linux and Java version 1.5.

For the decomposable graph search we recorded both the number of random proposals that resulted in a decomposable graph, and the number of these proposals that were accepted based on the usual Metropolis probabilities. For both types of interval graph searches we recorded the number of proposed new interval configurations that were accepted and also the number of these that resulted in a different implied graph. These counts are shown in Fig. 6. For all the versions of the program, the starting configuration used was the trivial graph, that is

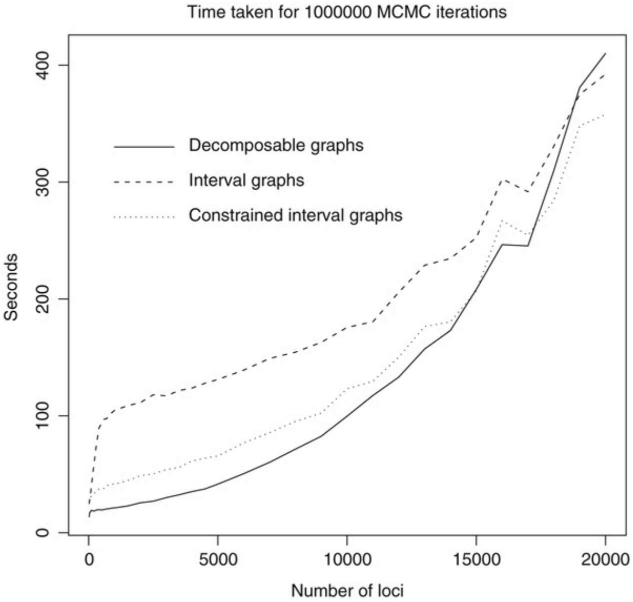


Fig. 4. The computer times required for one million MCMC iterations by number of genetic loci when the search is over general decomposable graphs, general interval graphs and constrained interval graphs.

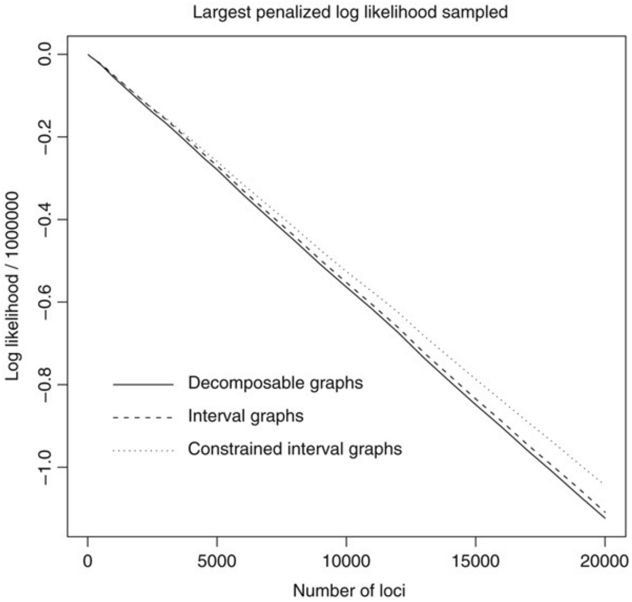
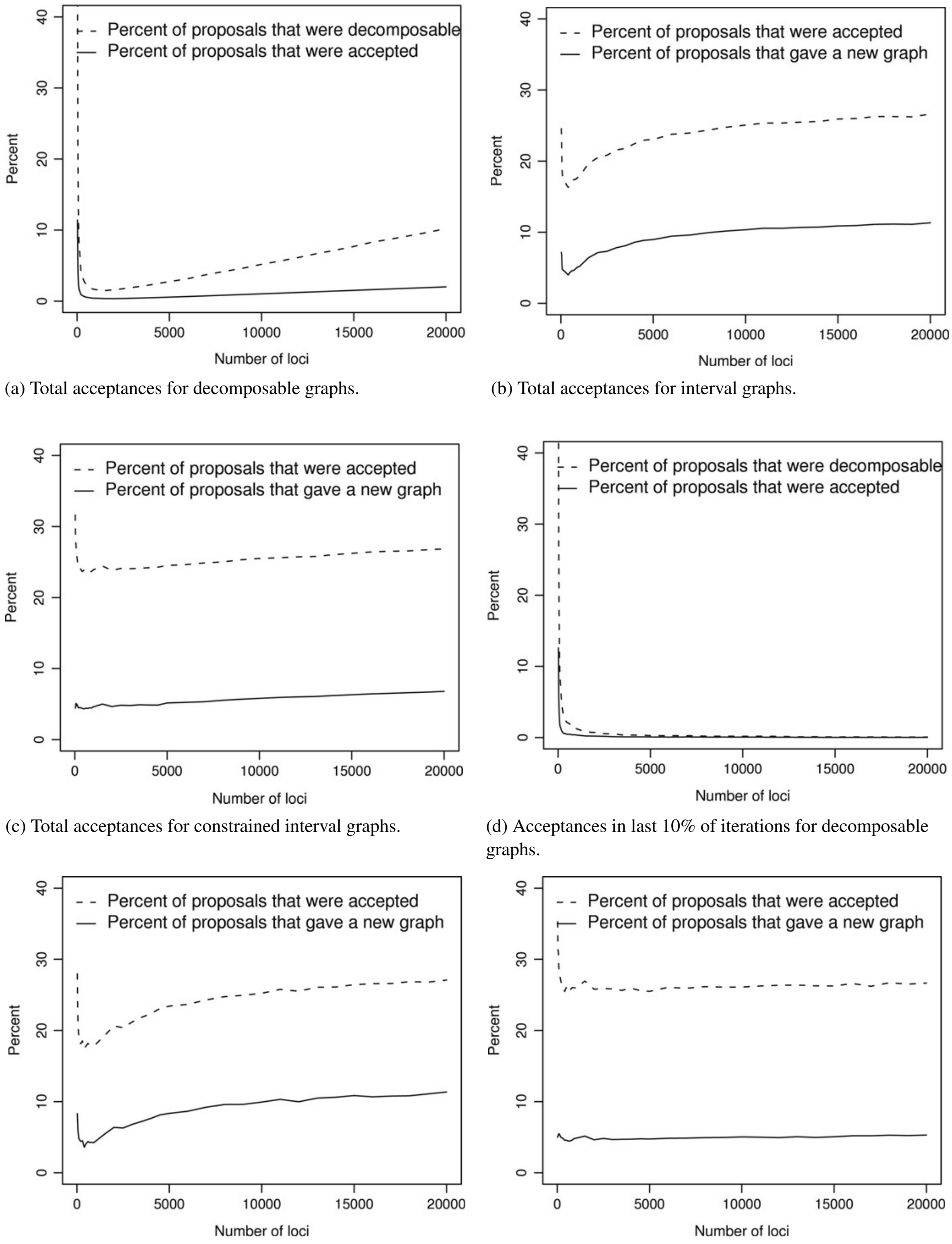


Fig. 5. The largest penalized log likelihood score seen in a sample of 1,000,000 MCMC simulations by number of loci.

the graph with a vertex for each locus but no edges. Thus, in the early stages of each search the graph was very sparse and almost all randomly chosen pairs of vertices could legitimately be connected to give a decomposable model. Also, in the early stages almost any change would tend to be accepted. Therefore, in order to check the performance of the methods closer to the equilibrium state we also recorded these counts in the last 100,000 (10%) of iterations. These are also shown in Fig. 6.

We then compared the haplotype frequencies implied by models optimized for each of the three classes of graph using simulated annealing. To avoid comparing very small frequencies we considered only haplotypes for the first 20 polymorphic loci on chromosome 1. Fig. 7 gives pairwise scatter plots of the frequencies estimated under the general decomposable models against those seen for general and constrained interval graphs. As an external reference we also show the haplotype frequencies estimated using the FASTPHASE program, those estimated with no



(a) Total acceptances for decomposable graphs.

(b) Total acceptances for interval graphs.

(c) Total acceptances for constrained interval graphs.

(d) Acceptances in last 10% of iterations for decomposable graphs.

(e) Acceptances in last 10% of iterations for interval graphs.

(f) Acceptances in last 10% of iterations for constrained interval graphs.

Fig. 6. The numbers of accepted proposals in all 1,000,000 MCMC simulations and in the final 100,000 simulations under the three classes of graphs considered by number of loci, shown as percentages.

accommodation for LD, that is, under the assumption of linkage equilibrium, and those estimated under the assumption that dependence is limited to a first order Markov chain and a fifth order Markov chain.

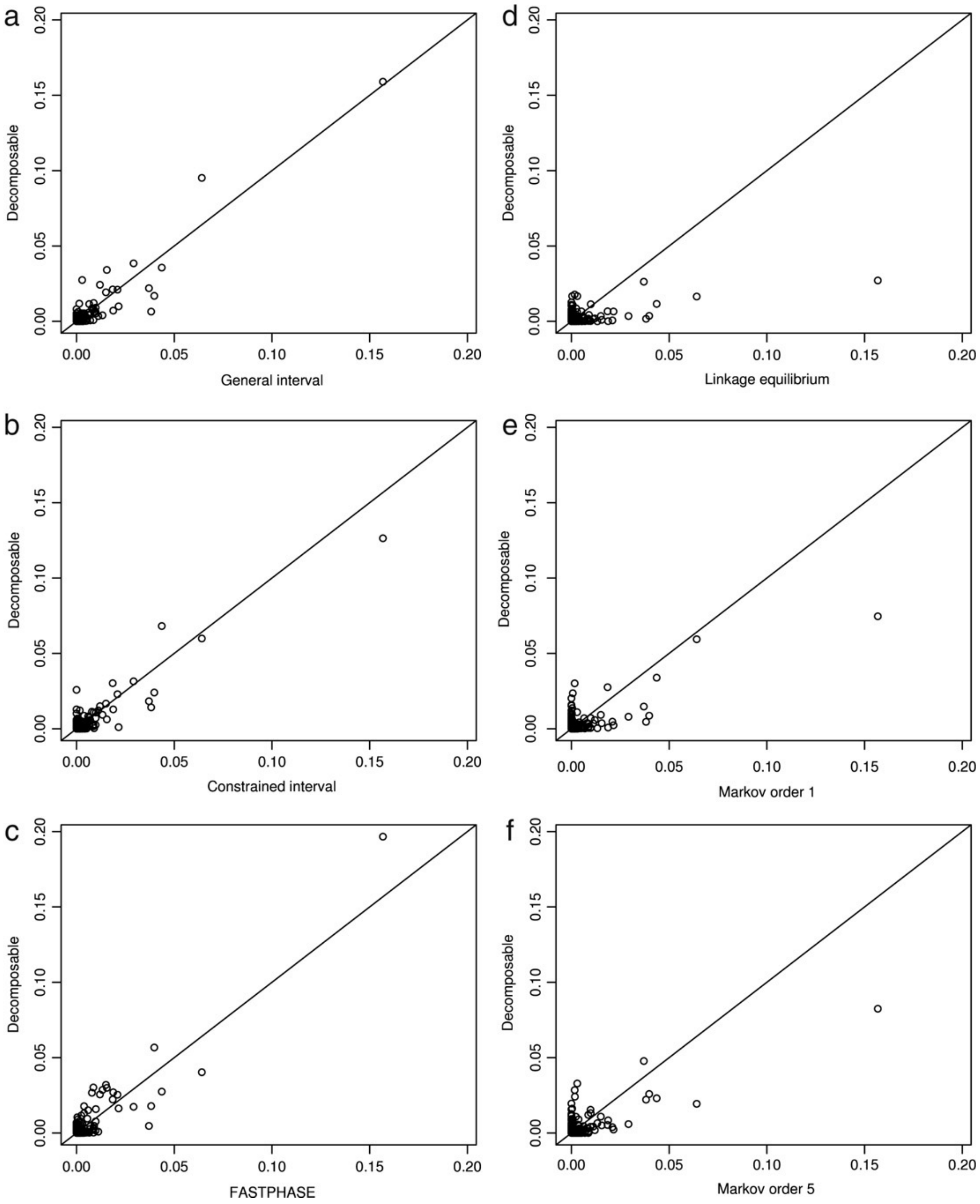


Fig. 7. Haplotype frequencies for the first 20 loci estimated for the YRI data for an optimized model with general decomposable graph compared with models with general interval and constrained interval graphs. Also compared are haplotype frequencies estimated using FASTPHASE, and those estimated under linkage equilibrium, and under first and fifth order Markov dependence.

4. Discussion

In absolute terms, as shown in Fig. 4, the computational performances for the three methods are similar. In the long run, the time required is quadratic although for up to about 10,000 variables performance is very close to linear. This difference is probably due to the increasing amounts of work done by the Java garbage collector to reclaim heap space. Even for the substantial numbers of loci used here, none of the methods takes prohibitive time or storage.

Although for below around 15,000 loci each of the interval graph methods takes more absolute time than the decomposable graph method, the amount of work done is substantially more as shown by Fig. 6. Fig. 6(b) and (c) show that around 25% of updates for the interval graph methods are accepted, of which 5% to 10% give rise to new graphs. For the decomposable graph method the percentage of proposals accepted decreases rapidly, see Fig. 6(a). The difference is far more marked in the last 100,000 iterations when the effect of initial conditions is minimized. Of the last 100,000 times that a random pair of 20,000 loci were selected, in only 70 cases could the pair be either disconnected, if they were previously connected, or connected, if they were previously disconnected, so that the resulting graph was decomposable: clearly the rejection method becomes very inefficient, see Fig. 6(d). On the other hand, for constrained interval graphs, the acceptance rate settles down very quickly at about 25%, and the accepted interval configurations that give a new graph settles at about 5%, Fig. 6(f).

The acceptance rate for general interval graphs actually increases with the number of loci, even for the last 100,000 iterations. However, this is likely to be due to long-term residual effects of initial conditions: in effect, for general interval graphs on large numbers of vertices the Markov chain is not mixing well. This poor mixing is also reflected in Fig. 5. Since constrained interval graphs are a subset of general interval graphs which are a subset of decomposable graphs, the true optimal values of the penalized log likelihood scores must increase through that sequence of inclusion. However, the maxima actually found reverse that order showing that the smaller space of constrained interval graphs is far more efficiently searched than its supersets.

The statistical effects of model subclassing are shown in Fig. 7. For this example the differences between haplotype frequencies estimated from models in each of the three classes of graphs are very similar, see Fig. 7(a) and (b). The results from FASTPHASE are also similar, Fig. 7(c). However, frequencies under linkage equilibrium or simple Markov dependence, even up to fifth order, show marked differences with far more points along or close to the axes of Fig. 7(d), (e) and (f). The distribution of distances between the 20 loci used here is quite skewed, with a mean of 37.97 kilo bases but median of only 2.33 kilo bases. Thus, haplotype frequencies derived from general decomposable, general interval, and constrained interval models are similar to each other and also similar to those derived from FASTPHASE. In contrast, ignoring LD completely or modelling it with small lag Markov models gives misleading results even though they may require more parameters than interval graphs.

Overall, therefore, there are clear computational benefits and little costs in terms of model flexibility to using interval graphs. In particular, in the context of LD modelling, constrained interval graphs have considerable practical advantages. As a final comment, note that the localization of the interactions implied in the constrained interval graph method means that loci sufficiently far apart can be considered separately. Thus, although not exploited by the programs described here, this would allow a moving window implementation that scales linearly with the number of loci and be feasible on a genome wide level.

Acknowledgments

This work was supported by grants number R01 GM070710 and R01 GM081417 to Alun Thomas from the National Institutes of General Medical Sciences. The content is solely the responsibility of the author and does not necessarily represent the official views of the National Institutes of General Medical Sciences or the National Institutes of Health.

References

Amos, C.I., Chen, W.V., Lee, A., Li, W., Kern, M., Lundsten, R., Batliwalla, F., Wener, M., Remmers, E., Kastner, D.A., Criswell, L.A., Seldina, M.F., Gregersen, P.K., 2006. High density SNP analysis of 642 Caucasian families with rheumatoid arthritis identifies two new linkage regions on 11p12 and 2q33. Genetic and Immunity 7, 277–286.

de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O., 2000. Compuational Geometry. Algroithms and Applications, second edn. Springer-Verlag.

Giudici, P., Green, P.J., 1999. Decomposable graphical Gaussian model determination. Biometrika 86, 785–801.

Golumbic, M.C., 1980. Algorithmic Graph Theory and Perfect Graphs. Academic Press.

Hastings, W.K., 1970. Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57 (1), 97–109.

Højsgaard, S., Thiesson, B., 1995. BIFROST — block recursive models induced from relevant knowledge, observations, and statistical techniques. Computational Statistics and Data Analysis 19, 155–175.

Kirkpatrick, S., Gellatt Jr., C.D., Vecchi, M.P., Optimization by simmulated annealing, Technical Report RC 9353, IBM, Yorktown Heights, 1982.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., 1953. Equations of state calculations by fast computing machines. Journal of Chemistry and Physics 21, 1087–1091.

Morton, N.E., 2002. Applications and extensions of Malecot's work in human genetics. In: Slatkin, M., Veuille, M. (Eds.), Modern Developments in Theoretical Population Genetics. Oxford University Press, Oxford, pp. 20–36.

Ott, J., 1985. Analysis of Human Genetic Linkage. The Johns Hopkins University Press, Baltimore.

Scheet, P., Stephens, M., 2006. A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase. American Journal of Human Genetics 78, 629–644.

Stephens, M., Smith, N.J., Donnelly, P., 2001. A new statistical method for haplotype reconstruction from population data. American Journal of Human Genetics 68, 978–989.

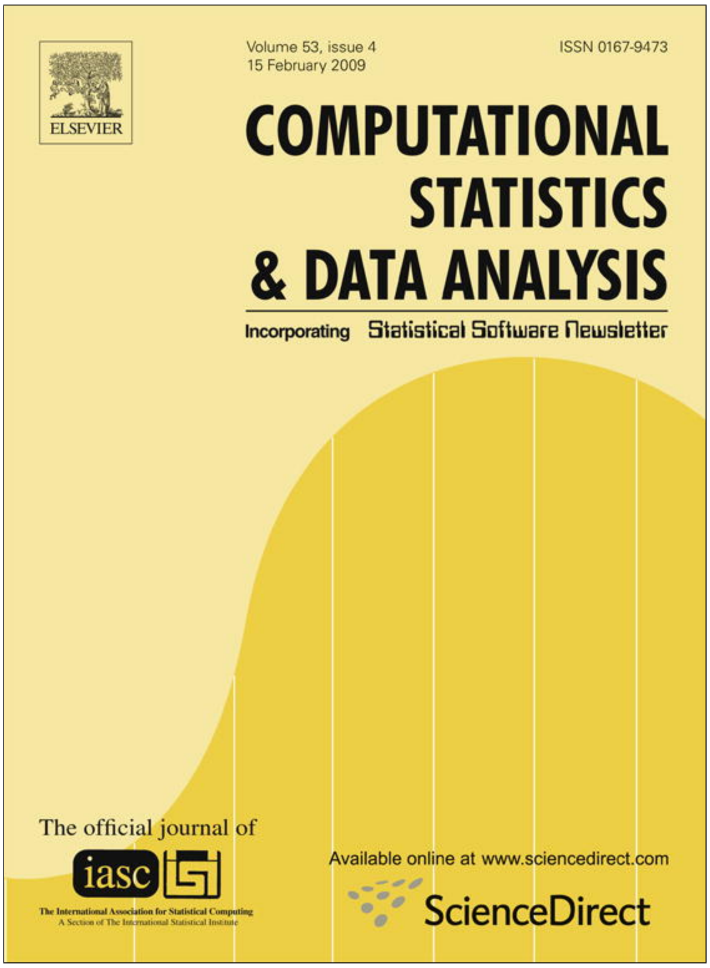
The International HapMap Consortium, 2005. A haplotype map of the human genome. Nature 437, 1299–1320.

Thomas, A., 2005. Characterizing allelic associations from unphased diploid data by graphical modeling. Genetic Epidemiology 29, 23–35.

Thomas, A., 2007. Towards linkage analysis with markers in linkage disequilibrium. Human Heredity 64, 16–26.

Thomas, A., Camp, N.J., 2004. Graphical modeling of the joint distribution of alleles at associated loci. American Journal of Human Genetics 74, 1088–1101.

Thomas, A., Camp, N.J., Farnham, J.M., Allen-Brady, K., Cannon-Albright, L.A., 2008. Shared genomic segment analysis. Mapping disease predisposition genes in extended pedigrees using SNP genotype assays. Annals of Human Genetics 72, 279–287.


ELSEVIER
Volume 53, issue 4
15 February 2009
ISSN 0167-9473
COMPUTATIONAL STATISTICS & DATA ANALYSIS
Incorporating Statistical Software Newsletter
The official journal of
iasc
The International Association for Statistical Computing
A Section of The International Statistical Institute
Available online at www.sciencedirect.com
ScienceDirect

Computational Statistics and Data Analysis 53 (2009) 1232–1238

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Enumerating the decomposable neighbors of a decomposable graph under a simple perturbation scheme

Alun Thomas [a,*], Peter J. Green [b]

[a] Department of Biomedical Informatics, University of Utah, USA

[b] Department of Mathematics, University of Bristol, Bristol BS8 1TW, UK

ARTICLE INFO

Article history:
Received 20 November 2007
Received in revised form 20 October 2008
Accepted 22 October 2008
Available online 5 November 2008

ABSTRACT

Given a decomposable graph, we characterize and enumerate the set of pairs of vertices whose connection or disconnection results in a new graph that is also decomposable. We discuss the relevance of these results to Markov chain Monte Carlo methods that sample or optimize over the space of decomposable graphical models according to probabilities determined by a posterior distribution given observed multivariate data.

1. Introduction

The flexibility and tractability of decomposable graphical models make them an attractive option for fitting to a broad array of complex multivariate data types. Commonly, the Markov, or conditional independence, graph of the graphical model is regarded as unknown, and is an objective of inference, along with parameters of the joint probability distribution of the data. This is the setting, of joint structural and parametric inference, that is considered here. Højsgaard and Thiesson (1995) developed a penalized likelihood approach to graphical model estimation that they implemented in the BIFROST program, which used a deterministic search method to find an optimal model. Other authors have used random methods to optimize the model, or sample from well fitting models, in effect implementing Metropolis–Hastings (Metropolis et al., 1953; Hastings, 1970) or simulated annealing (Kirkpatrick et al., 1982) samplers for the posterior probability distribution on decomposable graphical models corresponding to the penalized likelihood. Giudici and Green (1999) developed such methods for Gaussian models, while Thomas and Camp (2004) and Thomas (2005) applied this approach to modeling discrete distributions for allelic association between genetic loci in the same genomic region. The programs developed for this latter application can also be used to estimate a graphical model for general finite valued multivariate data. More recent work in graphical model estimation includes Jones et al. (2005) and Dobra et al. (2003).

Common algorithmic elements in these approaches are, given a decomposable graphical model with Markov graph G, to propose a new graph G' by either adding or deleting an edge, checking that the G' is decomposable, calculating its likelihood and prior probability, and hence either accepting or rejecting the proposal as appropriate according to the posterior probabilities. Implemented naively, checking for decomposability of G', using a maximum cardinality search (Tarjan and Yannakakis, 1984) for example, will take time of order $|V| + |E|$ where V and E are respectively the vertex and edge sets of G. Calculating the likelihood and prior of G' directly is also a substantial computation of order $|V|$. However, Giudici and Green (1999) showed that establishing the decomposability of G', given the decomposability of G, could be done in close to constant time when the perturbations involve either adding or deleting an edge and, moreover, that the difference in likelihood and prior between G and G' depends only on local differences that can be evaluated in time independent of

[*] Corresponding address: Genetic Epidemiology, 391 Chipeta Way Suite D, Salt Lake City, UT 84108, USA. Tel.: +1 801 587 9303; fax: +1 801 581 6052.
E-mail addresses: alun@genepi.med.utah.edu (A. Thomas), P.J.Green@bristol.ac.uk (P.J. Green).

0167-9473/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2008.10.029

the size of the graph. These results allow for much faster sampling and optimization methods even for moderately large graphs. For example, the HapGraph program (Thomas, 2005) for estimating graphical models for allelic association between genetic loci has been changed to use Giudici and Green's method and so works in reasonable time for several thousand variables.

However, for very large graphs the probability that the addition or deletion of a randomly proposed edge results in a decomposable graph becomes vanishingly small. Intuitively, for the relatively sparse graphical models often seen in practice, we would expect the number of allowable changes to be approximately linear in the size of the graph, whereas the number of all pairs is quadratic. Thus, as the size of the problem increases, any rejection algorithm becomes very inefficient. Recent advances in technology, particularly in molecular biology, allow the simultaneous measurement of hundreds of thousands of variables in a single assay, so, in this field at least, this low acceptance rate is a real obstacle to the use of graphical model based inference.

This problem could be avoided by characterizing and enumerating the set of pairs of vertices whose connection or disconnection results in a decomposable model, and then sampling a proposed perturbation with uniform probability from this set. Identifying the pairs whose disconnection results in a decomposable graph is easily done using Giudici and Green's result, and we illustrate this below. Identifying pairs which can be connected while preserving decomposability is more involved, but we develop here a framework that allows this and again provide an illustrative example.

2. Definitions and preliminary results

We begin by reviewing some definitions and standard properties of decomposable graphs and junction trees. A complete treatment of the topic is given by Lauritzen (1996).

Consider a graph $G = G(V, E)$ with vertices V and edges E. A subset of vertices $U \subseteq V$ defines an *induced subgraph* of G which contains all the vertices U and any edges in E that connect vertices in U. A subgraph induced by $U \subseteq V$ is *complete* if all pairs of vertices in U are connected in G. A *clique* is a complete subgraph that is maximal, that is, it is not a subgraph of any other complete subgraph.

Definition 1. A graph G is *decomposable* if and only if the set of cliques of G can be ordered as $(C_1, C_2, \dots, C_c)$ so that for each $i = 1, 2, \dots, c - 1$

$$\text{if } S_i = C_i \cap \bigcup_{j=i+1}^{c} C_j \quad \text{then } S_i \subset C_k \text{ for some } k > i. \tag{1}$$

This is called the *running intersection property*. Note that decomposable graphs are also known as *triangulated* or *chordal* graphs and that the running intersection property is equivalent to the requirement that every cycle of length 4 or more in G is chorded.

The sets $S_1, \dots, S_{c-1}$ are called the *separators* of the graph. The set of cliques $\{C_1, \dots, C_c\}$ and the collection of separators $\{S_1, \dots, S_{c-1}\}$ are uniquely determined from the structure of G, however, there may be many orderings that have the running intersection property. The cliques of G are distinct sets, but the separators are generally not all distinct. Let $S_{[1]}, \dots, S_{[s]}$ be the distinct sets contained in the collection of separators.

Definition 2. The *junction graph* of a decomposable graph has nodes $\{C_1, \dots, C_c\}$ and every pair of nodes is connected. Each link is associated with the intersection of the two cliques that it connects.

Note that for clarity we will reserve the terms *vertices* and *edges* for the elements of G, and call those of the junction graph and its subgraphs *nodes* and *links*.

Definition 3. Let J be any spanning tree of the junction graph. J has the *junction property* if for any two cliques C and D of G, every node on the unique path between C and D in J contains $C \cap D$. In this case J is said to be a *junction tree*.

For illustration, Fig. 1 gives an example of a decomposable graph while Fig. 2 shows one of its possible junction trees. Some authors first partition a graph into its disjoint components before making a junction tree for each component, combining the result into a *junction forest*. The above definition, however, will allow us to state results more simply without having to make special provision for nodes in separate components. In effect, we have taken a conventional junction forest and connected it into a tree by adding links between the components. Each of these new links will be associated with the empty set and have zero weight. Clearly, this tree has the junction property. Results for junction forests can easily be recovered from the results we present below for junction trees.

A junction tree for G will exist if and only if G is decomposable, and algorithms such as the *maximal cardinality search* of Tarjan and Yannakakis (1984) allow a junction tree representation to be found in time of order $|V| + |E|$. The collection of clique intersections associated with the $c - 1$ links of any junction tree of G is equal to the collection of separators of G. The junction property ensures that the subgraph of a junction tree induced by the set of cliques that contain any set $U \subseteq V$ is a single connected tree.

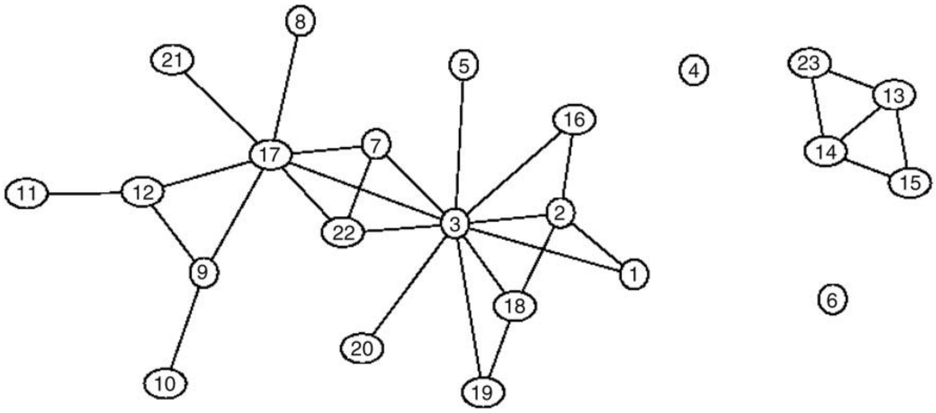


Fig. 1. A decomposable graph containing 23 vertices in 4 disjoint components.

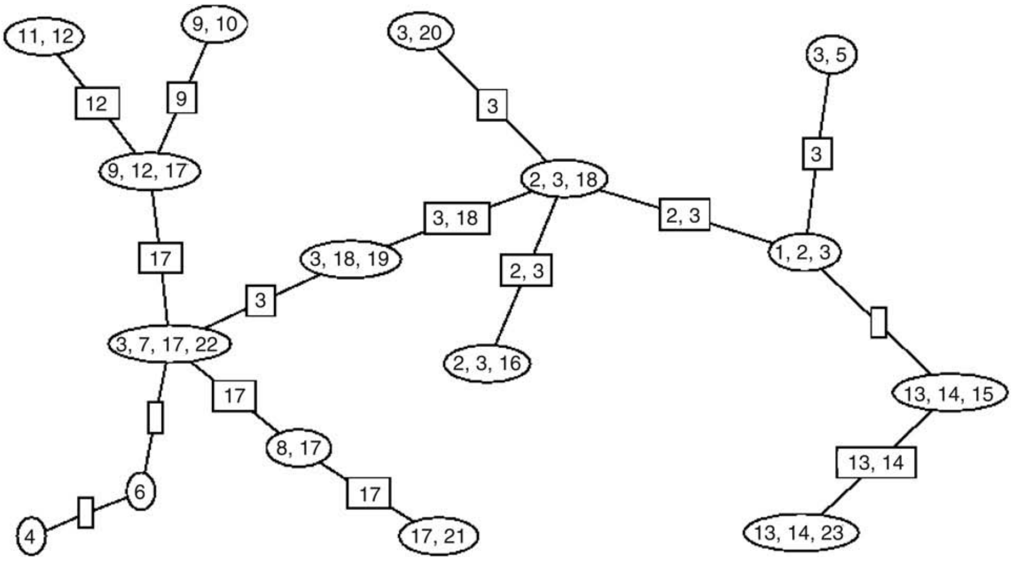


Fig. 2. One possible junction tree for the graph shown in Fig. 1. The 16 cliques are the vertices of the junction tree and are shown as ovals. The 15 clique separators are represented by the edges of the graph and each edge is associated with the intersection of its incident vertices. These intersections are shown as rectangles. Note that some of these intersections are empty.

3. Enumerating allowable perturbations

Frydenberg and Lauritzen (1989) and Giudici and Green (1999) gave efficient methods for checking that G' is decomposable, given that G is, when the perturbation scheme involves respectively disconnecting or connecting a random pair of vertices. Using our definition of a junction tree, we can restate their results as follows.

- Removing an edge (x, y) from G will result in a decomposable graph if and only if x and y are contained in exactly one clique.
- Adding an edge (x, y) to G will result in a decomposable graph if and only if x and y are unconnected and contained in cliques that are adjacent in some junction tree of G.

Thus, enumerating the disconnectible pairs of vertices is easy to do. Having found the cliques of G, we simply visit each in turn and count the number of edges that appear only in that clique. If C is the unique clique containing (x, y), we say that the disconnection of (x, y) is *allowed* by C.

Characterizing connectible pairs of vertices, however, is more complicated. If we keep track of the junction tree J representing our incumbent graph in the course of a sampling scheme, it is not enough to check the links of J to see whether x and y are in adjacent cliques. We also have to, in effect, consider all possible junction trees equivalent to J. To achieve this we first show that the set of connectible pairs can be partitioned over the distinct separators of G. We then give a method for identifying the pairs corresponding to each distinct separator.

Let x and y be unconnected vertices of decomposable G whose connection results in a new decomposable graph G'. Giudici and Green's condition ensures that G must have cliques $C_x \ni x$ and $C_y \ni y$ that are linked in some junction tree J of G. Let $S = C_x \cap C_y$ be the intersection associated with the link. Under this assumption the two following results hold.

Proposition 4. *There are no cliques $C'_x \ni x$ and $C'_y \ni y$ with either $C'_x \neq C_x$ or $C'_y \neq C_y$ that are adjacent in J.*

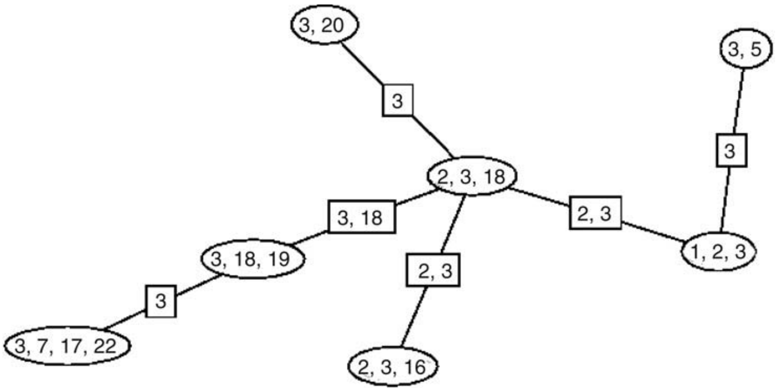


Fig. 3. $T_{\{3\}}$, the connected subtree of the junction graph shown in Fig. 2 induced by the cliques that contain the separator $\{3\}$.

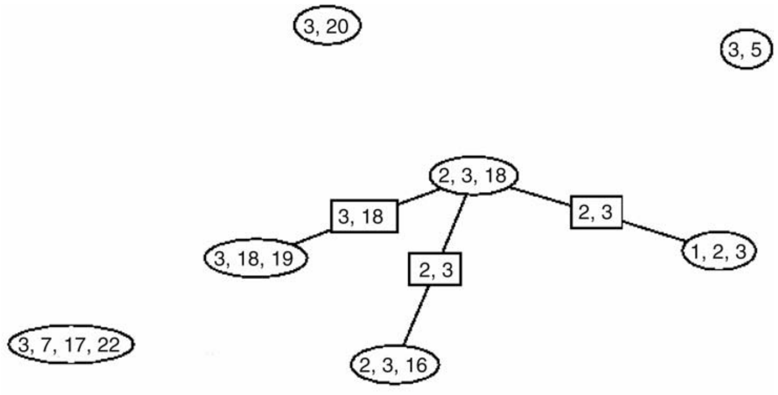


Fig. 4. $F_{\{3\}}$, the forest obtained by from the tree shown in Fig. 3 by deleting edges associated with the separator $\{3\}$.

Proof. Assume for the sake of contradiction that such C'_x and C'_y do exist. By the junction property there is a path from C'_x to C_x in J through nodes that all contain x, and similarly one from C'_y to C_y through nodes containing y. Since C_x and C_y are not both equal to C'_x and C'_y, at least one of these paths has length greater than 0. Also, since x and y are not connected they cannot appear in the same clique so these paths do not intersect. Thus, since C_x and C_y are adjacent, a link between C'_x and C'_y creates a cycle of at least 3 nodes violating J's tree property. □

Proposition 5. *Let $C'_x \ni x$ and $C'_y \ni y$ be cliques of G that are adjacent in some junction tree $J' \neq J$. Let $S' = C'_x \cap C'_y$. Then $S' = S$.*

Proof. Consider the positions of C'_x and C'_y in J. Again we construct the unique path in J from C'_x to C_x, C_x to C_y and C_y to C'_y, although it is possible now that $C'_x = C_x$ and $C'_y = C_y$ in which case the path has length 1. By the junction property S' must be contained in each node along the path, and in particular $S' \subseteq C_x$ and $S' \subseteq C_y$. Hence $S' \subseteq C_x \cap C_y = S$. By similarly considering the path from C_x to C_y in J' we also have that $S \subseteq S'$, and so $S' = S$. □

These two results immediately give us the following.

Theorem 6. *The set of all possible connectible pairs of vertices of a decomposable graph G can be partitioned according to the distinct separators of G. The subset of connectible pairs for a particular distinct separator S are those that appear in cliques adjacent in some junction tree and whose intersection is equal to S. We say that these pairs are allowed by S.*

Let T_S be the subtree of J induced by the cliques that contain the distinct separator S. The junction property ensures that T_S is a single connected subtree. For example, Fig. 3 shows $T_{\{3\}}$, the subtree of the junction tree in Fig. 2 defined by the separator $\{3\}$. Then, let F_S be the forest obtained from T_S by deleting all the links associated with S. For example, Fig. 4 shows the forest obtained by deleting links associated with the separator $\{3\}$ from $T_{\{3\}}$. If we now reconnect the subtrees of F_S by inserting copies of the separator S to make a new tree T'_S say, and then replace T_S in J by T'_S to make a new spanning tree J', J' will also be a junction tree of G because it spans all the cliques of G and has the same collection of separators associated with the links. Moreover, since F_S contains the only cliques of G that contain S, this represents all the ways in which the edges associated with S can be rearranged to make a new junction tree.

We can then characterize the pairs allowed by S as follows.

Theorem 7. *A connection (x, y) is allowed by a distinct separator S if and only if $x \notin S$, $y \notin S$ and x and y are in cliques that are nodes of different subtrees of F_S.*

Proof. We have established that the set of equivalent junction trees consists precisely of those obtained by inserting links associated with S into the gaps in F_S so as to form a tree. Then by Theorem 6, we only have to look at the cliques $(A \cup S, B \cup S)$ at either end of those links associated with S, and by Giudici and Green's result S allows connecting (x, y) if and only if $x \in A$ and $y \in B$ or vice versa. □

4. Method summary

It is already established that the set of disconnectible pairs of G can be partitioned among the cliques of G (Giudici and Green, 1999). Theorem 6 now establishes a corresponding result for the connectible pairs: that they can be partitioned among the distinct separators of G.

However, not all the edges in any given clique are disconnectible, only those that appear in no other cliques. The junction property makes this easy to check: if a pair of vertices (x, y) contained in clique C is also contained in another clique, they must appear in a neighbor of C in the junction tree, thus only the neighbors of C in J need to be checked. Let the number of pairs of disconnectible vertices allowed by C be $\beta(C)$.

Theorem 7 allows us to similarly find the connectible pairs allowed by any distinct separator S. These are pairs of vertices that do not appear in S and which do not appear in cliques in the same subtree of F_S. If we let m_S be the number of times a distinct separator S appears in the collection of all separators $\{S_1, \ldots, S_{c-1}\}$, since F_S is obtained by deleting m_s links from T_S, it must comprise $m_S + 1$ subtrees. Let A_j be the union of the sets of vertices in the cliques of the jth subtree of F_S. Let $a_j = |A_j - S|$, or $a_j = |A_j| - |S|$ since S must be contained in each A_j, and let $b = \sum_{i=j}^{m_S+1} a_j$. The number of connectible pairs allowed by S must therefore be

$$\alpha(S) = \frac{1}{2} \sum_{j=1}^{m_S+1} a_j(b - a_j).$$

Hence, an outline of a method to enumerate the neighborhood of a decomposable graph G is as follows:

- Given a decomposable graph G find any junction tree representation J.
- For each clique C_i of G:
  - For each pair of vertices (x, y) in C_i check that $\{x, y\}$ is not a subset of any neighboring clique of C_i in J.
  - $\beta(C_i)$ is the number of such pairs.
- The number of disconnectible edges in G is $\sum_i \beta(C_i)$.
- For each distinct separator $S_{[i]}$ of G:
  - Identify $T_{S_{[i]}}$ the connected subtree of J induced by the cliques containing $S_{[i]}$.
  - Find $J_{S_{[i]}}$ by deleting from $T_{S_{[i]}}$ the links corresponding to the separator $S_{[i]}$.
  - Calculate a_j for each of the $m_S + 1$ components of $F_{S_{[i]}}$.
  - Hence find $\alpha(S_{[i]}) = 1/2 \sum_j a_j(b - a_j)$.
- The number of connectible edges in G is $\sum_i \alpha(S_{[i]})$.

5. Illustration

As most of the above steps are either standard, such as finding a junction tree, or straightforward, such as checking that edges appear in only 1 clique, we illustrate only the novel enumeration of connectible edges. As an example, consider enumerating the connectible edges allowed in the graph in Fig. 1 by the separator $\{3\}$. The forest, $F_{\{3\}}$, shown in Fig. 4 has 4 components. The vertices in the cliques in each component, excluding vertex 3 are $\{20\}$, $\{7, 17, 22\}$, $\{5\}$ and $\{1, 2, 16, 18, 19\}$. Any pair of vertices selected from any two of these sets are connectible making

$$\alpha(\{3\}) = \frac{1}{2}(1 \times 9 + 3 \times 7 + 1 \times 9 + 5 \times 5) = 32$$

connections allowed by separator $\{3\}$. Note that the result of connecting the vertices 16 and 18 that appear in cliques in Fig. 4 is also a decomposable graph, however, this is allowed by the separator $\{2, 3\}$, not by $\{3\}$, so we do not count this possibility at this stage.

The other distinct separators are dealt with in the same way and the final results are presented in Table 2 which enumerates all the pairs of vertices in Fig. 1 whose connection makes a new decomposable graph. Table 1 shows an enumeration of the edges in Fig. 1 that are disconnectible. Of the ${}^{23}C_2 = 253$ possible pairs only $26 + 169 = 195$ can be changed to give a new decomposable graph.

6. Discussion

As noted above, a junction tree for a decomposable G can be found in time of order $|V| + |E|$. Letting $n = |V|$, this is at worst $O(n^2)$, and for the sparser graphs likely to be encountered in model estimation is closer to linear in n. As we build the junction tree, J, we can, at no extra cost, note for any distinct separator $S_{[i]}$ a clique in which it is contained. Thus, by searching out from that clique we can find $T_{S_{[i]}}$ in time proportional to the number of nodes in $T_{S_{[i]}}$. Since J can have at most n vertices, finding all the $T_{S_{[i]}}$ can be done in at most $O(n^2)$ time. While it is possible to construct examples where this bound is attained,

Table 1
An enumeration of the disconnectible pairs for the graph in Fig. 1.

Clique C	Edges only in C	Count
{4}	–	0
{6}	–	0
{13, 14, 15}	(13, 15) (14, 15)	2
{13, 14, 23}	(13, 23) (14, 23)	2
{3, 5}	(3, 5)	1
{1, 2, 3}	(1, 2) (1, 3)	2
{2, 3, 18}	(2, 18)	1
{2, 3, 16}	(2, 16) (3, 16)	2
{3, 20}	(3, 20)	1
{3, 18, 19}	(3, 19) (18, 19)	2
{17, 21}	(17, 21)	1
{8, 17}	(8, 17)	1
{9, 10}	(9, 10)	1
{11, 12}	(11, 12)	1
{9, 12, 17}	(9, 12) (9, 17) (12, 17)	3
{3, 7, 17, 22}	(3, 7) (3, 17) (3, 2) (7, 17) (7, 22) (17, 22)	6
Total number of disconnectible pairs		26

Table 2
An enumeration of the connectible pairs for the graph in Fig. 1.

Distinct separator S	m_S	Variables in cliques of components of F_S	Sizes	$\alpha(S)$

∅	4	{4}{6}{13, 14, 15, 23 }
		{1, 2, 3, 5, 7, 8, 9, 10, 11,
		12, 16, 17, 18, 19, 20, 21, 22}
			1, 1, 4, 17	111

{13, 14}	1	{15}{23}	1, 1	1
{3}	3	{20}{5}{7, 17, 22}{1, 2, 16, 18, 19}	1, 1, 3, 5	32
{2, 3}	2	{1}{16}{18 }	1, 1, 1	3
{3, 18}	1	{2}{19}	1, 1	1
{9}	1	{10}{12, 17}	1, 2	2
{12}	1	{11}{9, 17}	1, 2	2
{17}	3	{8}{21}{9, 12}{3, 7, 22}	1, 1, 2, 3	17
Total number of connectible pairs				169

for graphs encountered in model estimation actual performance is again likely to be far faster. Given $T_{S_{[i]}}$, with appropriate indexing of vertices, $J_{S_{[i]}}$ and $\alpha(S_{[i]})$ can be found in $O(n)$ time. Hence, overall, enumerating the connectible vertex pairs by our method allows an $O(n^2)$ worst case implementation.

The method previously described by Deshpande et al. (2001) also solves this problem in $O(n^2)$ time and is in many ways similar. It too, in essence, relies on an understanding of the result of Giudici and Green (1999) when stated as: *two unconnected vertices of G are connectible if and only if they are contained in cliques that are adjacent in some junction tree J of G*. In order to exploit this they require two auxiliary data structures that are both of order $O(n^2)$ in size. One of these, the *clique graph*, can be thought of as the union of all possible junction trees of G. It is here that our method has a significant advantage as it requires only a single, arbitrary, junction tree representation J of G. As J is of $O(n)$ size this represents a considerable saving of space. Moreover, as Giudici and Green (1999) showed, if G is updated to a decomposable G' by either the connection or disconnection of a pair of vertices, the corresponding junction tree J' can be found from J in $O(1)$ time. The update will depend on the context, but involves changes to 4 cliques or separators where a change may be the addition or deletion of the clique or separator. This compares very favorably with the $O(n^2)$ method for correspondingly updating the clique graph given by Deshpande et al. (2001).

Furthermore, adjustments to the probability of G to obtain that of proposed G' requires only the calculation of the clique marginals of the same 4 cliques or separators which is again quick and independent of the size of the graph. We also note that this does not require construction of G' so that both the proposal and acceptance/rejection steps of the Metropolis sampling process can be carried out using only the junction tree representation.

If we accept a proposal G', we could, at worst, then enumerate the allowable perturbations anew and select another random update. However, the structure of the junction tree enables the new enumeration to be made as a modification of the previous. Once more, counting the number of allowable disconnections is easy. We simply decrease the count by the number allowed in any clique removed, and increase by the number in any clique created.

When a link of J, associated with separator S, is added or removed, this will affect not only T_S but also T_U for any other separator $U \subset S$, so we need to track the partial ordering of the distinct separators defined by inclusion. The junction property can again help here as it ensures that if $U \subset S$ then T_S is a connected subgraph of F_U. Thus, the effect of changes to T_S on F_U, and hence the number of connections allowed by U, can be evaluated by searching in J starting at the changed link S and ending when, in each direction, a link associated with U is found.

There remains some challenge in developing specific algorithms and data structures to implement efficiently the methods outlined above. However, it is clear that the junction tree has very strong properties that can be exploited to make this possible.

Acknowledgments

This work was begun while the authors were participants at the Isaac Newton Institute for Mathematical Science's program on Stochastic Computation in the Biological Sciences in 2006 and we would like to thank the programme organizers, the Institute and its staff for their work and support.

This work was supported by NIH grants R21 GM070710 and R01 GM81417 to Alun Thomas.

References

Deshpande, A., Garofalakis, M.N., Jordan, M.I., 2001. Efficient stepwise selection in decomposable models. In: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence. pp. 128–135.

Dobra, A., Hones, B., Hans, C., Nevins, J., West, M., 2003. Sparse graphical models for exploring gene expression data. Journal of Multivariate Analysis 90, 196–212.

Frydenberg, M., Lauritzen, S.L., 1989. Decomposition of maximum likelihood in mixed interaction models. Biometrika 76, 539–555.

Giudici, P., Green, P.J., 1999. Decomposable graphical Gaussian model determination. Biometrika 86, 785–801.

Hastings, W.K., 1970. Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57 (1), 97–109.

Højsgaard, S., Thiesson, B., 1995. BIFROST — Block recursive models Induced From Relevant knowledge, Observations, and Statistical Techniques. Computational Statistics and Data Analysis 19, 155–175.

Jones, B., Carvalho, C., Dobra, A., Hans, C., Carter, anc C., West, M., 2005. Experiments in stochastic computation for high-dimensional graphical models. Statistical Science 20, 388–400.

Kirkpatrick, S., Gellatt, C.D. Jr., Vecchi, M.P., 1982. Optimization by simulated annealing. Technical Report RC 9353, IBM, Yorktown Heights.

Lauritzen, S.L., 1996. Graphical Models. Clarendon Press.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., 1953. Equations of state calculations by fast computing machines. Journal of Chemistry and Physics 21, 1087–1091.

Tarjan, R.E, Yannakakis, M., 1984. Simple linear-time algorithms to test the chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. SIAM Journal of Computing 13, 566–579.

Thomas, A., 2005. Characterizing allelic associations from unphased diploid data by graphical modeling. Genetic Epidemiology 29, 23–35.

Thomas, A., Camp, N.J, 2004. Graphical modeling of the joint distribution of alleles at associated loci. American Journal of Human Genetics 74, 1088–1101.

Enumerating the junction trees of a decomposable graph

Alun Thomas[*]
Department of Biomedical Informatics
University of Utah

Peter J Green[†]
Department of Mathematics
University of Bristol

May 7, 2009

Keywords: Chordal graphs, triangulated graphs, graphical models, Markov chain Monte Carlo methods.

Running title: Enumerating junction trees

Abstract

We derive methods for enumerating the distinct junction tree representations for any given decomposable graph. We discuss the relevance of the method to estimating conditional independence graphs of graphical models and give an algorithm that, given a junction tree, will generate uniformly at random a tree from the set of those that represent the same graph. Programs implementing these methods are included as supplemental material.

[*] Genetic Epidemiology, 391 Chipeta Way Suite D, Salt Lake City, UT 84108, USA. alun@genepi.med.utah.edu, +1 801 587 9303 (voice), +1 801 581 6052 (fax).

[†] Department of Mathematics, University of Bristol, Bristol BS8 1TW, UK, P.J.Green@bristol.ac.uk.

1 Introduction

Decomposable or *triangulated* or *chordal* graphs are of interest in many areas of mathematics. Our primary interest is in their role as the conditional independence graphs of decomposable graphical models. In particular, we are interested in estimating decomposable graphical models from observed data using Markov chain Monte Carlo, or *MCMC*, schemes that traverse the space of decomposable graphs in order to sample from, or maximize, the posterior probability distribution defined by the data. The underlying approach is the same whether the data is continuous (Giudici & Green 1999, Jones et al. 2005) or discrete (Thomas & Camp 2004, Thomas 2005). A common feature of such schemes is that, given an incumbent decomposable graph G, we propose a new graph G' which is then accepted or rejected according to probabilities that depend on the distribution being sampled (Metropolis et al. 1953, Hastings 1970, Kirkpatrick et al. 1982). However, there are no known proposal schemes that guarantee in advance that G' will be decomposable, even if G is. Hence, it is necessary to test G' for decomposability before evaluating the usual acceptance probability. While such tests can be very quick (Giudici & Green 1999), for all practical methods for proposing a random G' of which we are aware, the probability that G' is decomposable decreases rapidly with the size of the graph, making this approach infeasible for large problems. For instance, in the genetic examples considered by Thomas (2009), involving up to 20,000 variables, the probability of proposing a decomposable G' decreases roughly as the inverse of the number of variables. Given these circumstances, it would be very useful to have an alternative representation of the problem that avoids the test for decomposability. It is with this in mind that we consider what follows.

It is often convenient in graphical modeling to operate not on the graph itself, but on its derived representation as a *junction tree*. This raises the prospect of discarding the underlying conditional independence graph entirely and defining MCMC schemes on the space of junction trees. As each junction tree uniquely defines a decomposable graph, this might avoid the expensive need to propose non-decomposable models. However, decomposable graphs have multiple equivalent junction tree representations and moreover the number is variable from graph to graph. Therefore, sampling the space of junction trees will over-sample decomposable graphs with a large number of such representations. This can be corrected for if the number of junction trees for any particular decomposable graph can be evaluated and this is the motivation for the method we present here.

We begin by reviewing some definitions and standard properties of decomposable graphs and junction trees. For more complete information on these see Golumbic (1980) and Lauritzen (1996), whose terminology we have adopted. We then consider the number of ways that sets of links of a junction tree that correspond to the same clique intersection can be rearranged. These counts are then combined to give the total number of junction trees. A simple algorithm is then presented that will take a junction tree and select an equivalent one uniformly at random from the set of all possible equivalents. Finally, we discuss the computational complexity of our method showing that it is faster than existing algorithms, and outline potential junction tree sampling methods.

2 Definitions and preliminary results

Consider a graph $G = G(V, E)$ with vertices V and edges E. A subset of vertices $U \subseteq V$ defines an *induced subgraph* of G which contains all the vertices U and any edges in E that connect vertices in U. A subgraph induced by $U \subseteq V$ is *complete* if all pairs of vertices in U are connected in G. A *clique* is a complete subgraph that is maximal, that is, it is not a subgraph of any other complete subgraph of G.

Definition 1 *A graph G is* decomposable *if and only if the set of cliques of G can be ordered as $(C_1, C_2, \ldots, C_c)$ so that for each $i = 1, 2, \ldots, c-1$*

$$\text{if } \; S_i \; = \; C_i \cap \bigcup_{j=i+1}^{c} C_j \; \text{ then } \; S_i \subset C_k \; \text{ for some } \; k > i. \tag{1}$$

This is called the *running intersection property* and is equivalent to the requirement that every cycle in G of length 4 or more is chorded. The sets $S_1, \ldots S_{c-1}$ are called the *separators* of the graph. The set of cliques $\{C_1, \ldots C_c\}$ and the collection of separators $\{S_1, \ldots S_{c-1}\}$ are uniquely determined from the structure of G; however, there may be many orderings that have the running intersection property. The cliques of G are distinct sets, but the separators are generally not all distinct.

Definition 2 *The* junction graph *of a decomposable graph has nodes $\{C_1, \ldots, C_c\}$ and every pair of nodes is connected. Each link is associated with the intersection of the two cliques that it connects, and has a weight, possibly zero, equal to the cardinality of the intersection.*

Note that for clarity we will reserve the terms *vertices* and *edges* for the elements of G, and call those of the junction graph and its subgraphs *nodes* and *links*.

Definition 3 *Let J be any spanning tree of the junction graph. J has the* junction property *if for any two cliques C and D of G, every node on the unique path between C and D in J contains $C \cap D$. In this case J is said to be a* junction tree.

Figure 1 gives an example of a decomposable graph while Figure 2 shows one of its possible junction trees. The lexicographic search method of Tarjan & Yannakakis (1984) will find a junction tree for a given decomposable graph in time and storage of order $|V| + |E|$.

Note that some authors first partition a graph into its disjoint components before making a junction tree for each component, combining the result into a *junction forest*. The above definition, however, will allow us to state results more simply without having to make special provision for nodes in separate components. In effect, we have taken a conventional junction forest and connected it into a tree by adding links between the components. Each of these new links will be associated with the empty set and have zero weight. Clearly, this

Figure 1: A decomposable graph containing 23 vertices in 4 disjoint components.

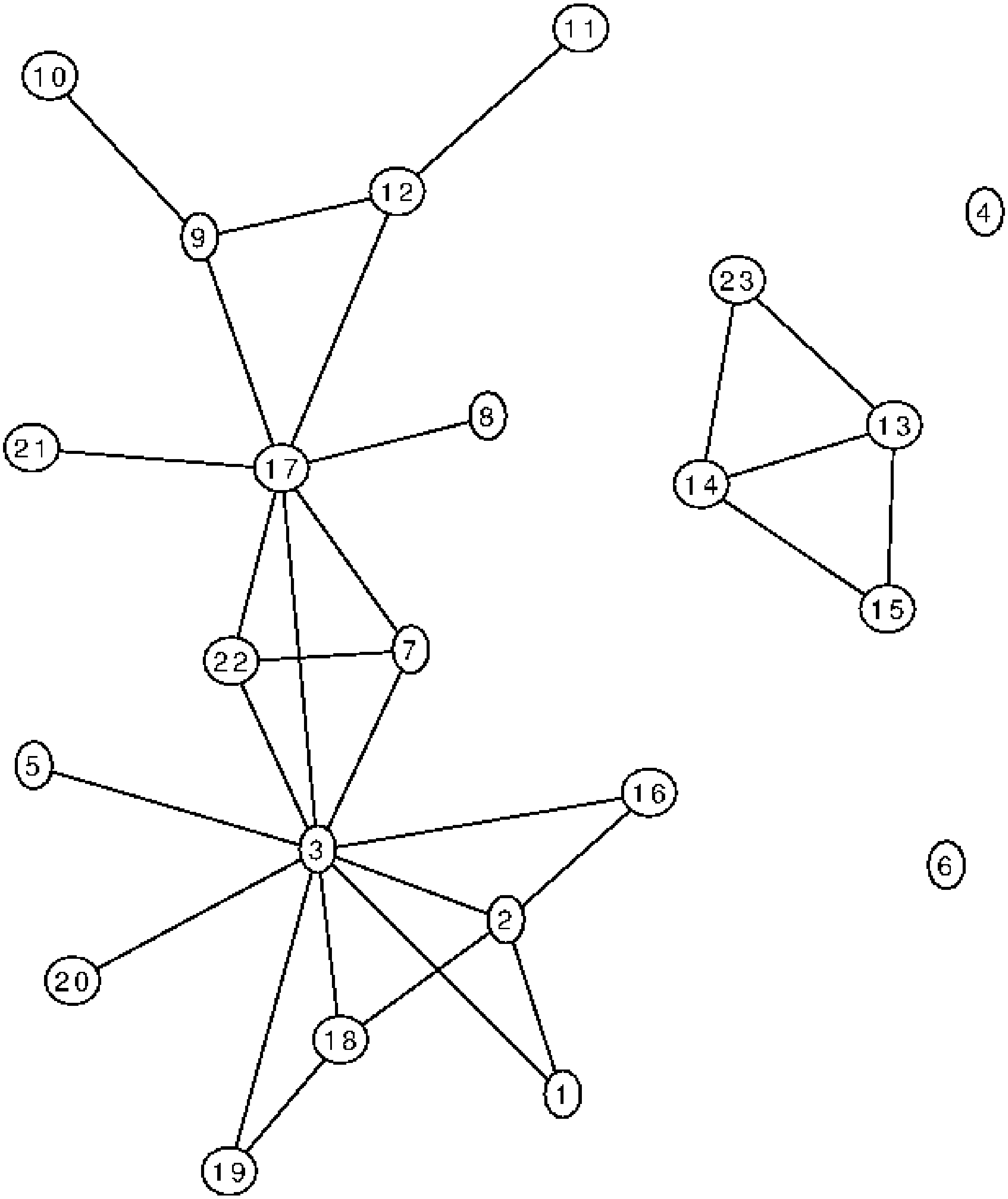
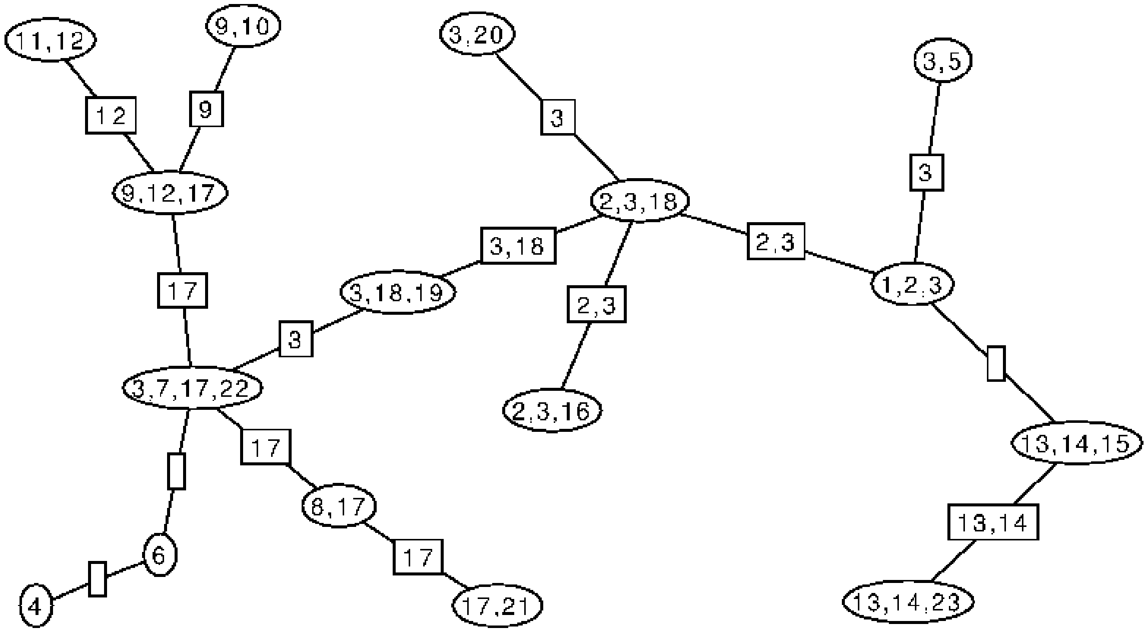


Figure 2: One possible junction tree for the graph shown in Figure 1. The 16 cliques are the vertices of the junction tree and are shown as ovals. The 15 clique separators are represented by the edges of the graph and each edge is associated with the intersection of its incident vertices. These intersections are shown as rectangles. Note that some of these intersections are empty.



tree has the junction property. Results for junction forests can easily be recovered from the results we present below for junction trees.

As Lauritzen (1996) describes more fully, a junction tree for G will exist if and only if G is decomposable, and the collection of clique intersections associated with the $c-1$ links of any junction tree of G is equal to the collection of separators of G. Also, the junction property can be equivalently stated as the property that the subgraph of a junction tree induced by the set of cliques that contain any set $U \subseteq V$ is a single connected tree.

As stated in Definition 2, we can consider each link of the junction graph to have a weight. Thus, any subgraph of it, and in particular any spanning tree, can also be associated with a weight defined by the sum of the weights of the links included. Jensen (1988) exploits this to give the following useful characterization of a junction tree.

Theorem 4 *A spanning tree of the junction graph is a junction tree if and only if it has maximal weight.*

From this it is clear that any tree with the cliques of G as its nodes and for which the collection of clique intersections associated with the links is equal to the collection of separators of G, is a junction tree of G, since such a tree must span the junction graph and have maximal weight. Therefore, the problem of enumerating junction trees for a given graph G is equivalent to enumerating the ways that links of a given junction tree can be rearranged so that the result is also a tree, and the collection of clique intersections defined by the links of the tree is unchanged.

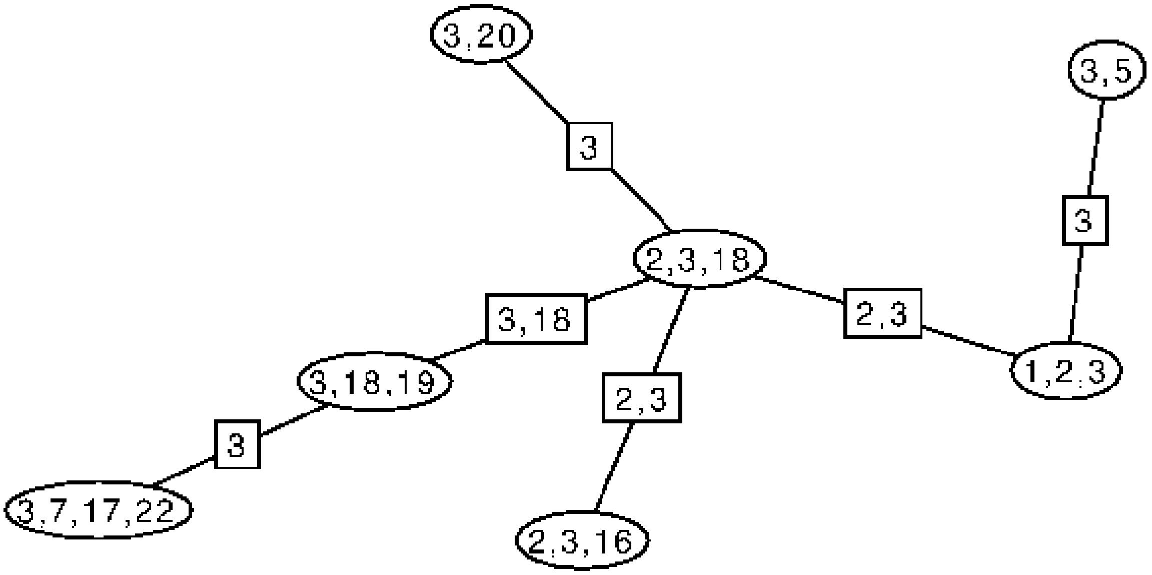
3 Rearranging the links for the set of separators with the same intersection

As noted above, the separators of G are not generally distinct. For example, in Figure 2 three links are associated with the clique intersection $\{17\}$ and two with the intersection $\{2, 3\}$. We now consider the effect of rearranging all the links that are associated with the same clique intersection. Let J be any junction tree of G and S one of its separators. Define T_S to be the subtree of J induced by the cliques that contain S. The junction property ensures that T_S is a single connected subtree of J.

Clearly, any rearrangement of the links associated with S in J must be a rearrangement among certain links of T_S, since both cliques joined by such a link must contain S. For illustration, Figure 3 shows $T_{\{3\}}$, the subtree defined by the separator $\{3\}$ for the graph in Figure 1. If we now rearrange the links that are associated with S to produce a new graph, T'_S say, and replace T_S in J by T'_S to give a new graph J', J' will be a junction tree of G if and only if

- T'_S is a tree, and hence so is J', and
- T'_S has the same weight as T_S, so that J' has the same weight as J.

Figure 3: $T_{\{3\}}$, the connected subtree of the junction graph shown in Figure 2 induced by the cliques that contain the separator $\{3\}$.



In fact the second condition is redundant: all cliques in T_S contain S so their intersection must also do so, and any pair of cliques whose intersection is a superset of S cannot be joined in a tree T'_S unless already joined in T_S as T'_S would then have greater weight than T_S, and J' greater weight than J thus violating the latter's maximal weight property. So we need only count the number of ways of rearranging the links of T_S associated with S such that T'_S is a tree.

Consider F_S to be the forest obtained by deleting all the links associated with S from T_S. For example, Figure 4 shows $F_{\{3\}}$, the forest obtained by deleting links associated with the separator $\{3\}$ from the tree $T_{\{3\}}$ shown in Figure 3. Define $\nu(S)$ to be the number of ways that the components of F_S can be connected into a single tree by adding the appropriate number of links. This value is given by a theorem by Moon (1970) which can be restated as follows.

Theorem 5 *The number of distinct ways that a forest of p nodes comprising q subtrees of sizes $r_1 \dots r_q$ can be connected into a single tree by adding $q-1$ links is*

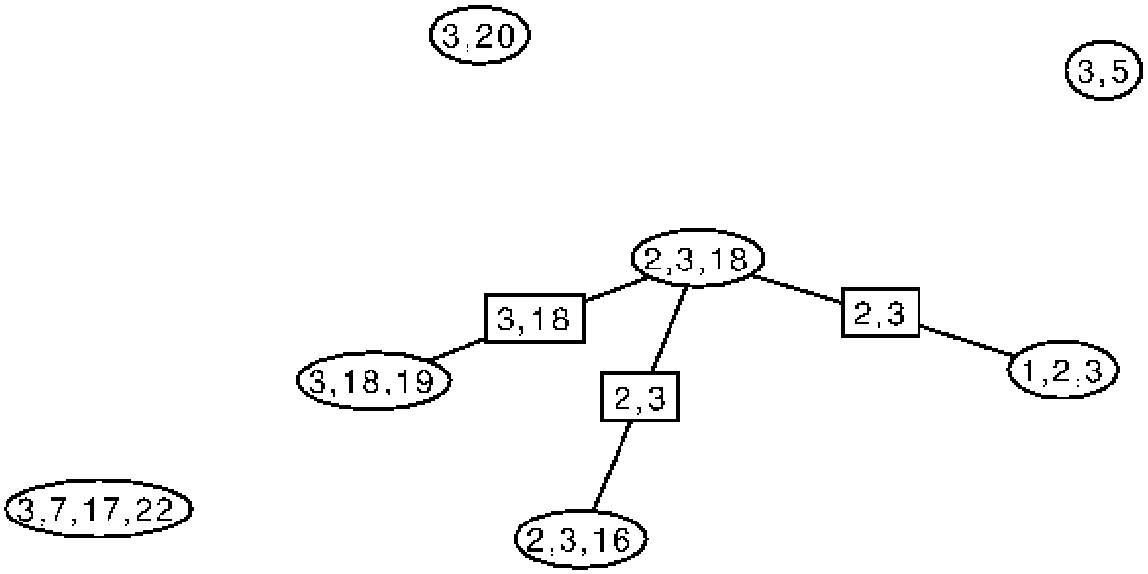
$$p^{q-2} \prod_{i=1}^{q} r_i. \tag{2}$$

If the number of links associated with a given separator S is m_S we know that F_S will contain $m_S + 1$ components. Let these be of sizes $f_1, f_2, \dots f_{m_S+1}$. Let the number of nodes in T_S be t_S which is simply the number of cliques of G that contain S. Then, directly from theorem 5 we obtain the following.

Theorem 6

$$\nu(S) = t_S^{m_S-1} \prod_{j=1}^{m_S+1} f_j. \tag{3}$$

Figure 4: $F_{\{3\}}$, the forest obtained by from the tree shown in Figure 3 by deleting edges associated with the separator $\{3\}$.



For example, the forest in Figure 4 has 7 nodes in 4 components of sizes $1, 1, 1$ and 4. This forest, $F_{\{3\}}$, can be reconnected into a single tree by adding 3 links in $7^2 \times 1 \times 1 \times 1 \times 4 = 196$ different ways.

4 The number of junction trees for a decomposable graph

The final step in enumerating junction trees is to note that $\nu(S)$ depends only on the sizes of the components of F_S, not on their particular structure. These sizes are determined by the sets of cliques that contain separators that are supersets of S. Since the set of cliques and collection of separators are uniquely determined and independent of any particular junction tree, $\nu(S)$ is independent of J. Hence, the links associated with one particular separator can be reallocated independently of the links associated with another. Thus we obtain the following result.

Table 1: The computations that enumerate the distinct junction trees for the decomposable graph given in Figure 1.

Separator S	m_S	t_S	$\{f_S\}$	$\nu(S)$
$\emptyset$	3	16	1, 1, 2, 12	6144
$\{13, 14\}$	1	2	1, 1	1
$\{3\}$	3	7	1, 1, 1, 4	196
$\{2, 3\}$	2	3	1, 1, 1	3
$\{3, 18\}$	1	2	1, 1	1
$\{9\}$	1	2	1, 1	1
$\{12\}$	1	2	1, 1	1
$\{17\}$	3	4	1, 1, 1, 1	16

$$\mu(G) = 6144 \times 1 \times 196 \times 3 \times 1 \times 1 \times 1 \times 16 = 57802752$$

Theorem 7 *Consider a decomposable graph G with separators $S_1, \ldots S_{c-1}$. Let $S_{[1]}, \ldots S_{[s]}$ be the distinct sets contained in the collection of separators. The number of junction trees of G is*

$$\mu(G) = \prod_{i=1}^{s} \nu(S_{[i]}). \tag{4}$$

As an example, the number of distinct junction trees for the graph shown in Figure 1 is 57,802,752. The calculations needed to obtain this are given in Table 1.

As noted above, we can retrieve from this result the count of the number of possible conventional junction forests that a decomposable graph has. This is given simply by

$$\frac{\mu(G)}{\nu(\emptyset)},$$

which for the example is $57802752/6144 = 9408$.

5 Randomizing the junction tree

Theorem 5 is similar in style to Prüfer's constructive proof (Prüfer 1918) of Cayley's result that there are n^{n-2} distinct labeled trees of n vertices (Cayley 1889). A similar construction lets us choose uniformly at random from the set of possible trees containing a given forest as follows:

1. Label each vertex of the forest $\{i, j\}$ where $1 \leq i \leq q$ and $1 \leq j \leq r_i$, so that the first index indicates the subtree the vertex belongs to and the second reflects some ordering within the subtree. The orderings of the subtrees and of vertices within subtrees are arbitrary.

2. Construct a list v containing $q-2$ vertices each chosen at random with replacement from the set of all p vertices.

3. Construct a set w containing q vertices, one chosen at random from each subtree.

4. Find in w the vertex x with the largest first index that does not appear as a first index of any vertex in v. Since the length of v is 2 less than the size of w there must always be at least 2 such vertices.

5. Connect x to y, the vertex at the head of the list v.

6. Remove x from the set w, and delete y from the head of the list v.

7. Repeat from 4 until v is empty. At this point w contains 2 vertices. Connect them.

Given any particular junction tree representation J for a decomposable graph G we can choose uniformly at random from the set of equivalent junction trees by applying the above algorithm to each of the forests F_S defined by the distinct separators of J.

6 Computational complexity

Jensen's characterization of a junction tree as a maximal spanning tree of the junction graph means that general methods for enumerating the optimal spanning trees of a graph can also be applied to enumerating junction trees. The method of Broder & Mayr (1997) does precisely this. Recalling the notation used in section 2, the junction graph will have c nodes and $c(c-1)/2$ links. Broder and Mayr's method would require $O(M(c))$ elementary operations to enumerate its maximal spanning trees, where $M(c)$ is the number of operations needed to multiply $c \times c$ matrices. Asymptotically, the best algorithm for matrix multiplication is that of Coppersmith & Winograd (1990) which requires $O(c^{2.376})$ operations, although for realistically sized matrices the best practical methods, based on that of Strassen (1969), need $O(c^{2.807})$ operations. Letting $n = |V|$, the number of vertices in G, we note that c can be as large as n and typically grows linearly with n. Hence, Broder and Mayr's algorithm is at best an $O(n^{2.376})$ method.

However, as noted above, Jensen's characterization is not the only route to obtaining a junction tree. The lexicographic search of Tarjan & Yannakakis (1984) will find a simple elimination scheme, and hence a junction tree, in time $O(n+m)$, where $m = |E|$ the number of edges in G. While $m = O(n^2)$ in the worst case, typical graphical models are sparse and the time required is closer to linear in n. The enumeration method presented here depends only on knowing a single junction tree for G. The time required to carry it out is dominated by the time needed to find each $T_{S_{[i]}}$. We note that any link L of J will be a link in $T_{S_{[i]}}$ for each i such that $S_{[i]} \subseteq L$. Finding all the $T_{S_{[i]}}$ can be done by iterating over the $c-1$ links of J, and for each link checking for inclusion of each of the s distinct separators. Since both c and s can be $O(n)$, the enumeration is an $O(n^2)$ algorithm in the worst case. Other ways of finding the $T_{S_{[i]}}$ will in practice be faster. For example, we can find $T_{S_{[i]}}$ by starting with a node that contains $S_{[i]}$ and searching outwards in J until we hit nodes that don't contain the separator. Thus, if $T_{S_{[i]}}$ is small it will be found very quickly. While it is straightforward to construct examples where this approach is also $O(n^2)$, for more typical graphs it will be considerably faster.

In summary, applying Broder and Mayr's general method to the special case of enumerating junction trees is at best an $O(n^{2.376})$ method, and more realistically $O(n^{2.807})$. By exploiting the junction property, our method improves this to a worst case of $O(n^2)$ which in practice is a very conservative upper bound.

7 MCMC samplers for junction trees

Given a distribution $\pi(G)$ from which we want to sample decomposable graphs G, the methods of Metropolis et al. (1953) and Hastings (1970) allow us to construct Markov chains with $\pi(G)$ as the ergodic distribution. For example, we can propose a new graph G' by choosing two random vertices of G: if they are connected in G we disconnect them in G' and vice versa. G' is then accepted with probability $\max(1, \pi(G')/\pi(G))$, with the special case that $\pi(G')$ is defined to be 0 if G' is not a decomposable graph. Intuitively, it is easy to see that this can be very inefficient. If we consider choosing two random vertices of G, it is quite likely that we pick two vertices that are not connected directly, but which are connected by several paths through other vertices. Adding an edge between the vertices is, therefore, likely to create cycles. Unless all the connecting paths are short, a cycle of length 4 or more may well be formed which prevents G' being decomposable. Thomas (2009) shows that for modeling linkage disequilibrium between genetic loci, the acceptance rate decreases approximately as $1/n$, making the method infeasible for large numbers of variables. As genetic methods now routinely assay hundreds of thousands of loci on a single chromosome, the high rejection rate becomes increasingly problematic.

The motivation for our enumeration method is that it makes it possible to devise MCMC schemes over decomposable graphs that are expressed as operations on junction trees. If we wish to sample decomposable graphs from $\pi(G)$, it is sufficient to sample junction trees from

$$\rho(J) = \frac{\pi(G(J))}{\mu(G(J))}$$

since

$$\begin{aligned} P(G) &= \sum_{J:G(J)=G} \frac{\pi(G(J))}{\mu(G(J))} \\ &= \frac{\pi(G)}{\mu(G)} \sum_{J:G(J)=G} 1 \\ &= \pi(G). \end{aligned}$$

For each J sampled from a Markov chain with ergodic distribution $\rho(J)$, we would derive the graph $G(J)$ which would be sampled with probability $\pi(G)$, as required. This, of course, requires efficient enumeration, but note that the Metropolis-Hastings acceptance probability for a junction tree MCMC scheme depends only on $\mu(G(J'))/\mu(G(J))$ which, given the factorization in equation 4 above, might be computable from $\mu(G(J))$ more simply than direct enumeration of $\mu(G(J'))$.

Simulating general labeled trees is a relatively straightforward matter. For instance, Prüfer's construction (Prüfer 1918) makes independent realizations of trees of a given size from a uniform distribution easy. However, for the junction tree problem the labels on the nodes of J are the cliques of G, and these must be connected so that the junction property holds, making for a more difficult problem in a constrained space. Nonetheless, we have been able to construct an irreducible MCMC sampling scheme over the space of junction trees for graphs of a given size n. This involves operations on the nodes and links of an incumbent junction tree J that correspond to either adding edges to or deleting edges from G when the edges are chosen from highly restricted sets of possibilities. Following these perturbation rules ensures that any proposal J' is a junction tree for some decomposable graph G' on n vertices. Hence, we avoid both the need to test for decomposability and the inefficiency of proposing non-decomposable graphs. The randomization method described in section 5 above can also be included in the scheme; although, it is not necessary for irreducibility, it may improve the mixing properties of the chain. While the space of junction trees is larger than that of decomposable graphs, it is more tractable and may be more easily traversible. A complete description of the method and implementation is the subject of a future manuscript currently under preparation.

8 Acknowledgments

This work was begun while the authors were participants at the Isaac Newton Institute for Mathematical Science's Programme on Stochastic Computation in the Biological Sciences in 2006 and we would like to thank the program organizers, the Institute and its staff for their work and support.

We would also like to thank three anonymous reviewers and the journal editors for their prompt and constructive comments on how to better present and motivate this work.

This work was supported by grants NIH R21 GM070710, NIH R01 GM81417, and DOD W81XWH-07-01-0483 to Alun Thomas.

9 Supplemental material

The following materials are available to accompany this paper.

`jtree.jar`: This file contains the source code and compiled classes required to run the programs described below. The code can be run by adding `jtree.jar` to the list of files accessed by the user's classpath, or by using the `-classpath` option of the `java` command. (Jar, java archive, file.)

docs: This is a directory of HTML pages automatically produced for the code in `jtree.jar` using the `javadoc` program, and can be viewed using a web browser. (Directory of HTML files.)

`illustration`: This is simple text file specifying the graph used in this paper to illustrate junction tree enumeration. (Text file.)

The programs included in `jtree.jar` are:

`CountJTrees`: This reads a graph as a simple text file from the standard input stream and outputs the number of equivalent junction trees it has. If the input graph is not decomposable, zero is output. For example, to replicate the enumeration shown in Table 1, first ensure that files `jtenum.jar` and `illustration` are in the current working directory and then run the command:

```
java -classpath jtenum.jar CountJTrees < illustration
```

The following programs can be run in a similar way.

`CountLogJTrees`: This reads a graph as a simple text file and outputs the log of the number of equivalent junction trees it has. If the input graph is not decomposable, negative infinity is output.

`FindRandomJTree`: This reads a graph and outputs a junction tree selected uniformly at random from all possible junction tree representations. If the input graph is not decomposable, there is no output.

`RandomJTreeDemo`: This reads a graph and draws a junction tree representation to the screen. Every 2 seconds the junction tree is replaced by randomly chosen equivalent junction tree. If the input graph is not decomposable, the program exits with an error message.

References

Broder, A. Z. & Mayr, E. W. (1997), Counting minimum weight spanning trees, *Journal of Algorithms* **24**, 171–176.

Cayley, A. (1889), A theorem on trees, *Quarterly Journal of Mathematics* **23**, 376–378.

Coppersmith, D. & Winograd, S. (1990), Matrix multiplication via arithmetic progressions, *Journal of Symbolic Computation* **9**, 251–280.

Giudici, P. & Green, P. J. (1999), Decomposable graphical Gaussian model determination, *Biometrika* **86**, 785–801.

Golumbic, M. C. (1980), *Algorithmic Graph Theory and Perfect Graphs*, Academic Press.

Hastings, W. K. (1970), Monte Carlo sampling methods using Markov chains and their applications, *Biometrika* **57**(1), 97–109.

Jensen, F. V. (1988), Junction trees and deomposable hypergraphs, Technical report, Judex Datasystemer, Aalborg, Denmark.

Jones, B., Carvalho, C., Dobra, A., Hans, C., Carter, C. & West, M. (2005), Experiments in stochastic compuation for high-dimensional graphical models, *Statistical Science* **20**, 388–400.

Kirkpatrick, S., Gellatt, Jr., C. D. & Vecchi, M. P. (1982), Optimization by simmulated annealing, Technical Report RC 9353, IBM, Yorktown Heights.

Lauritzen, S. L. (1996), *Graphical Models*, Clarendon Press.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N. & Teller, A. H. (1953), Equations of state calculations by fast computing machines, *Journal of Chemistry and Physics* **21**, 1087–1091.

Moon, J. W. (1970), Enumerating labelled trees, *in* F. Harary, ed., Graph Theory and Theoretical Physics, Academic Press, London.

Prüfer, H. (1918), Neuer beweis eines satzes uber permutationen, *Archiv fur Mathematik und Physik* **27**, 142–144.

Strassen, V. (1969), Gaussian elimination is not optimal, **13**, 354–356.

Tarjan, R. E. & Yannakakis, M. (1984), Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs, *SIAM Journal of Computing* **13**, 566–579.

Thomas, A. (2005), Characterizing allelic associations from unphased diploid data by graphical modeling, *Genetic Epidemiology* **29**, 23–35.

Thomas, A. (2009), Estimation of graphical models whose conditional independence graphs are interval graphs and its application to modeling linkage disequilibrium, *Computational Statistics and Data Analysis* **53**, 1818–1828.

Thomas, A. & Camp, N. J. (2004), Graphical modeling of the joint distribution of alleles at associated loci, *American Journal of Human Genetics* **74**, 1088–1101.

Anomalous shared genomic segments in high risk cancer pedigrees and HapMap control data.

Zheng Cai[12]

and

Kristina Allen-Brady

and

Alun Thomas

Received ______________________; accepted ______________________

Abstract

The method of shared genomic segment analysis[1] has shown that long tracts of loci that share a common allele identical by state can be used to localize hypothesized predisposition genes by indicating underlying regions shared identical by descent among individuals with a common trait. In this study, we found that there are long tracts of loci on chromosomes 5 and 18 at which both familial cancer cases from Utah pedigrees and European control individuals from HapMap share a common allele. These two, and only these two, regions stand out from the background levels of sharing in a genome wide analysis. They are not shared in African and Asian controls. These anomalous regions of heterozygous sharing are peculiar to Europeans, and perhaps to the Utah population, and emphasize the need for appropriate modeling in any simulation methods used to evaluate statistical significance.

Keywords: Identity by descent, extended pedigrees, association study, pedigree analysis, single nucleotide polymorphism (SNP).

Running title: Anomalous genomic sharing

1. INTRODUCTION

Identifying genes that have underlying susceptibilities to human genetic diseases represents a major challenge in biomedical research. While family based methods have

[1] Affiliation: Department of Biomedical Informatics, University of Utah.

[2] Corresponding author. Contact Information: Genetic Epidemiology, 391 Chipeta Way Suite D, Salt Lake City, UT 84108, USA. z.cai@utah.edu, +1 801 587 9303 (voice), +1 801 581 6052 (fax).

proved successful in the past using linkage analysis, this method is not tractable in the face of large sets of single nucleotide polymorphisms, or SNPs, that can currently be assayed. However, this type of data has been used in familial studies through methods of mapping by identifying regions of identity by descent, or IBD. Mapping homozygous IBD regions was introduced by Miyazawa et al[2] and can be applied to searching for recessive diseases in inbred populations. Thomas et al[1] developed this idea in extended pedigrees, searching for regions shared in a heterozygous state by relatives. This uses tracts of loci in which sets of individuals share a common allele to indicate underlying IBD. Leibon et al[3] and Kong et al[4] developed the same idea. We call this approach *genetic mapping by shared genomic segments.* While not as powerful as full linkage analysis, this method provides a robust and tractable alternative. By obtaining the largest number of individuals who share an allele at each marker, we search for an excessively long tract of unphased haplotype. Such a haplotype that is shared by all individuals is then taken as evidence that there is an underlying genomic segment inherited IBD from a common ancestor. This IBD region consequently becomes a candidate for containing a gene with a mutation causing susceptibility to the disease. Tracts of sharing among large subsets of individuals can also be considered. The statistical significance of long shared genomic segments can be evaluated by gene drop simulation. While the same idea was independently published by Leibon et al[3], they evaluated significance by extending methods derived by Miyazawa et al[2].

Both the gene drop approach of Thomas et al[1] and the distributions derived by Leibon et al[3] assume that the genetic loci are in linkage equilibrium. This simplifying assumption is clearly inappropriate for dense SNP maps, as the above authors pointed out. Linkage disequilibrium will likely increase the lengths of random shared genomic segments under the null distribution of no genetic cause for the disease.

The focus of this paper will be to describe the two long shared genomic segments, seen on chromosome 5 and 18. These two segments appear to be statistical anomalies in that they are well outside the range of variation seen elsewhere in a genome wide scan. They were originally detected in a set of individuals with prostate cancer from extended Utah pedigrees. This gave us the initial impression that genes for prostate cancer might be located in these regions. However, precicely the same shared segments were seen in an analysis of a set of melanoma cases. Moreover, these shared genomic regions were also seen in a set of 60 unrelated northwestern European descendent controls genotyped by the Hapmap project[5]. Clearly, that such anomalies can occur in unselected controls could potentially lead to spurious conclusions, and this paper seeks to describe and explain them. This emphasizes the potential pitfalls of naive modeling of haplotype frequencies in any analysis.

In what follows we will briefly review the method of genetic mapping by shared genomic segments describe the data analyzed and present the results of the analyses. Finally, we will discuss the possible causes of the anomalies. Programs for the analyses are available from Alun Thomas's web site (http://bioinformatics.med.uath.edu/∼alun).

2. SUBJECTS AND METHODS

2.1. Case and control sets

Using the Illumina 550K assay of over half a million SNP loci, we genotyped two sets of familial disease clusters, one of prostate cancer and one of melanoma, identified using the Utah Population Database (UPDB), a computerized genealogy database that has been record linked to cancer registry data. The prostate cancer set consisted of two large extended families. The first of eight distantly related cases connected by 27 meioses to a married pair of ancestors, the second of 21 distantly related cases connected by 68 meioses to a common ancestor pair. The melanoma set had more cases, 90, but distributed in 21 smaller pedigrees. Shared genomic segment analyses were performed on the individual pedigrees, as well as the combined disease sets. The data for the combined disease sets are the most relevant and presented below.

As controls for these disease cases, we took genotypes of 60 parents from the 30 parent-offspring trios of northern and western European origin genotyped by the HapMap project (Version r23a.b36 reversed-strand non-redundant data). These samples, conventionally denoted as CEU, were originally collected by the Centre d'Etude du Polimorphisme Humain in Utah in 1980, thus they should be well matched with cancer cases from the UPDB. We found that 95% of the autosomal markers genotyped in our assays were also genotyped in these individuals. The results presented below are from this intersection of $531,924$ markers.

In following up our findings we also used the CEU controls to measure the heterozygosity at each marker using the usual statistic of $1 - \hat{p}_{i,1}^2 - \hat{p}_{i,2}^2$ where $\hat{p}_{i,1}$ and $\hat{p}_{i,2}$ are the maximum likelihood estimates of the frequencies of alleles 1 and 2 at the ith locus. We also computed Chi-squared test statistics for Hardy-Weinberg equilibrium[6–8] at each marker. Linkage disequilibrium in the anomalous regions was also evaluated from the CEU data using the Haploview software[9] with the following criteria: ignore pairwise comparisons of markers more than 500Kb apart, take Hardy-Weinberg p-value cutoff of 0.001, and minimum minor allele frequency of 0.001.

Finally, in order to assess whether the results seen were specific to Europeans, we performed shared segment analyses on the HapMap data, using the genotypes of 60 parents out of 30 YRI trios, which are Yoruba people from Ibadan, Nigeria. We also tested the genotypes of 45 unrelated Han Chinese individuals from Beijing (CHB), and the 45 unrelated Japanese individuals from Tokyo (JPT); as well as combing the two as an Asian data set.

2.2. Shared genomic segments

Consider a genotyping assay of s SNPs carried out on n individuals. Define $n_{1,1}$ $n_{1,2}$ and $n_{2,2}$ as the counts of the genotypes at the ith locus. Note that $n_{1,1} + n_{1,2} + n_{2,2} \leq n$, with inequality when there are missing genotypes. Define $S_i = n - \min(n_{1,1}, n_{2,2})$, which is the largest number of individuals who could possibly share an allele provided there is no genotyping error. Any missing individuals are effectively treated as heterozygotes. We then compute the lengths of consecutive loci at which $S_i \geq t$ for some chosen values of the threshold t. Defining another variable $R_i(t)$, as the length of the longest tract containing the ith locus for which $S_i \geq t$. Leibon et al[3] called this the *SNP streak* statistic.

3. RESULTS

Figure 1 gives the plots for the tract lengths where (a) all 29 prostate cancer cases, (b) all 90 melanoma cases, and (c) all 119 combined cases, shared a common allele. For each of the diseases, the longest tracts were at 5q22.1 and 18q22.1. Also, as is demonstrated in the combined plot, these regions corresponded exactly across the two case sets. The shared segment on chromosome 5 spans 70 SNP markers from base pair positions 109,641,683 to 110,171,067, at a length of 529 Kb. That on chromosome 18, the shared segment spans 55 SNP markers and is 107 Kb in length from base pair positions of $64,802,946$ to $64,909,997$.

Figure 2 shows the tracts where (a) all of the 60 CEU individuals, (b) 59 out of the 60 CEU individuals, and (c) all 60 YRI individuals, share an allele. The first figure showed that there was again sharing of the segment on chromosome 18, but the sharing was not seen in all 60 individuals at chromosome 5. However, there were two longer than average tracts of sharing adjacent to each other at 5q22.1, and, as the second figure shows, if we relax the criterion to require all but one of the individuals to share, then the region on chromosome 5 again stands out. The non-sharing individual had a miss match at only one locus, which may be a true miss match or possibly a genotyping error.

Figure 3 shows the tracts where (a) all 45 CHB individuals, (b) all 45 JPT individuals, (c) all 90 combined CHB and JPT individuals from HapMap, share an allele.

Figure 4 gives the empirical distribution of heterozygosity scores across all of chromosomes 5 and 18, compared with the distribution of scores seen in the anomalous regions. Figure 5 similarly demonstrates the empirical distribution of Hardy-Weinberg test statistics over the whole chromosomes, as well as at the two anomalous regions. Finally, figure 6 and 7 shows plots of the strength of pairwise linkage disequilibrium scores between the genotyped markers, in and around the regions on chromosome 5 and 18.

4. DISCUSSION

The existence of two such extreme outliers in the otherwise rather even distribution of shared genomic segments in European case and control data is quite striking. Note also that the sharing in these regions is heterozygous, that is, that only one chromosome consistently appears to be shared. There are heterozygous genotypes observed at loci over both regions in several individuals. The anomalies are not seen in the YRI, JPT and CHB controls. There is a noticeable region of sharing on chromosome 5 among the Asian controls, see figure 3, but this does not overlap at all with that seen in the Europeans. There are no outstanding regions of sharing in the Africans (figure 2), indeed the tracts of sharing in this sample are considerably shorter than those seen in the others as would be expected given the greater variation and lower linkage disequilibrium seen in Africa[10]. While the CEU controls are appropriate for our Utah genetic studies, we are unable to determine whether the anomalies are specific to the northern European background of the Utah founders, or extend more broadly in Europe. Note, however, that the large number of Utah founders and the average levels of inbreeding seen in the Utah population, meaning that the outliers are unlikely to be due to founder effects or genetic bottle necks. Note also that they are not due to high levels of missing data. The missing data rate across the whole CEU data was 1%, and only 0.7% and 0.4% in the chromosome 5 and 18 regions respectively. Furthermore, the allele frequencies, as shown in figure 5 appear to be in Hardy-Weinberg equilibrium.

Our results do, however, suggest some possible explanations. The region at 18q22.1 is, perhaps, the easier to explain. It is a shorter segment and matches almost exactly a linkage disequilibrium block, or recombination cold spot, as shown in figure 7. Moreover, this region generally has low levels of heterozygosity, as seen in figure 4. As the minor allele will be rare at loci in the region, and observations at successive loci will be strongly correlated, we may simply have missed seeing rare homozygotes due to random sampling of SNPs. Since a tract of allele sharing is usually ended by the appearance of the rarer homozygote, the apparent shared segment may be due to the combination of these two features.

The 5q22.1 region, on the other hand, is longer, and has higher heterozygosity. Also, it does not have a particularly strong linkage disequilibrium structure. However, there is a copy number variant, reported by Redon et al[11], between $109,669,760$ and $110,180,038$bp, closesly matching the anomalous shared segment. Even though its association with any gene function is currently unknown, it is asserted to be a copy number variant with one-copy loss. If the anomaly were due to a common deletion among Europeans, we would expect to see deviations from Hardy-Weinberg equilibrium due to an apparent excess of homozygotes, however, as noted above we do not see this. A supernumerary copy number variant, on the other hand, could lead to excess apparent heterozygotes which might result in excessive apparent allele sharing.

An alternative explanation for the chromosome 5q22.1 phenomenon would be positive selection. There are large negative values of Tajima's D statistic[12] throughout most of this region, as calculated using the UCSC genome browser[13–15]. This indicates an excess of rare variation in the CEU population, which is consistent with positive selection. This region is supported by data obtained from Haplotter[10]. There are both strongly negative iHS score ($|iHS| > 2.5$) and strongly negative values of Fay and Wu's H score within the region ($|H| > 2.0$) for CEU samples, suggesting that this is a highly selected site[16]. Moreover, the 5q22.1 region contains a putative uncharacterized protein *FLJ43080*[17], and a gene *SLC25A46*[18;19]. *SLC25A46* is a member of the mitochondrial solute carrier family that is widely expressed in the central nervous system. High expression has been detected in the hindbrain, coronal section IV, spinal cord, and section VII[19]. Studies[10;20;21] have suggested that such genes involved in brain development and function are likely targets for natural selection in recent human evolution. The location of *SLC25A46* and the statistical survey suggest that this anomaly is due to recent positive selection.

Whatever the causes of these anomalies, they demonstrate the need for complex modeling of haplotype frequencies in genetic studies. Even methods that account for linkage disequilibrium may not properly capture the full range of allelic association that may make itself apparent.

5. Acknowledgment

Data collection for this publication was supported by Grant Number R01 GM081417 from the National Institutes of General Medical Sciences. The content is solely the responsibility of the author(s) and does not necessarily represent the official views of the

National Institutes of General Medical Sciences or the National Institutes of Health. Data collection for this publication was supported by US Army Medical Research and Material Command (USAMRMC), W81XWH-07-1-0483 (to Alun Thomas). Research was supported by the Utah Cancer Registry, which is funded by contract N01-PC-35141 from the National Cancer Institute's SEER program with additional support from the Utah State Department of Health and the University of Utah. The pedigrees in this analysis were developed using the Utah Population Database (UPDB). Partial support for all datasets within the Utah Population Data Base was provided by the University of Utah Huntsman Cancer Institute.

The authors are grateful to the discussion with Dr. Ellen Wijsman, Dr. Frederick Adler and Dr. Jon Seger.

REFERENCES

Thomas, A., Camp, N., Farnham, J., Allen-Brady, K., and Cannon-Albright, L. (2008). Shared Genomic Segment Analysis. Mapping Disease Predisposition Genes in Extended Pedigrees Using SNP Genotype Assays. Annals of Human Genetics *72*, 279–287.

Miyazawa, H., Kato, M., Awata, T., Kohda, M., Iwasa, H., Koyama, N., Tanaka, T., Kyo, S., Okazaki, Y., and Hagiwara, K. (2007). Homozygosity Haplotype Allows a Genomewide Search for the Autosomal Segments Shared among Patients. The American Journal of Human Genetics *80*, 1090–1102.

Leibon, G., Rockmore, D., and Pollak, M. (2008). A SNP Streak Model for the Identification of Genetic Regions Identical-by-descent. Statistical Applications in Genetics and Molecular Biology *7*, 16.

Kong, A., Masson, G., Frigge, M., Gylfason, A., Zusmanovich, P., Thorleifsson, G., Olason, P., Ingason, A., Steinberg, S., Rafnar, T., et al. (2008). Detection of sharing by descent, long-range phasing and haplotype imputation. Nature Genetics *40*, 1068.

Gibbs, R., Belmont, J., Hardenbol, P., Willis, T., Yu, F., Yang, H., Ch'ang, L., Huang, W., Liu, B., Shen, Y., et al. (2003). The International HapMap Project. Nature *426*, 789–796.

Hardy, G. (1908). MENDELIAN PROPORTIONS IN A MIXED POPULATION.

Weinberg, W. (1908). Über den Nachweis der Vererbung beim Menschen. Jahreshefte des Vereins für vaterländische Naturkunde in Württemberg *64*, 368–382.

STERN, C. (1943). THE HARDY-WEINBERG LAW. Science *97*, 137–138.

Barrett, J., Fry, B., Maller, J., and Daly, M. (2005). Haploview: analysis and visualization of LD and haplotype maps.

Voight, B., Kudaravalli, S., Wen, X., and Pritchard, J. (2006). A Map of Recent Positive Selection in the Human Genome. PLoS Biol *4*, e72.

Redon, R., Ishikawa, S., Fitch, K., Feuk, L., Perry, G., Andrews, T., Fiegler, H., Shapero, M., Carson, A., Chen, W., et al. (2006). Global variation in copy number in the human genome. Nature *444*, 444–454.

Tajima, F. (1989). Statistical Method for Testing the Neutral Mutation Hypothesis by DNA Polymorphism. Genetics *123*, 585–595.

Kent, W., Sugnet, C., Furey, T., Roskin, K., Pringle, T., Zahler, A., and Haussler, D. (2002). The Human Genome Browser at UCSC.

Karolchik, D., Kuhn, R., Baertsch, R., Barber, G., Clawson, H., Diekhans, M., Giardine, B., Harte, R., Hinrichs, A., Hsu, F., et al. (2008). The UCSC Genome Browser Database: 2008 update. Nucleic Acids Research *36*, D773.

Carlson, C., Thomas, D., Eberle, M., Swanson, J., Livingston, R., Rieder, M., and Nickerson, D. (2005). Genomic regions exhibiting positive selection identified from dense genotype data.

Sabeti, P., Schaffner, S., Fry, B., Lohmueller, J., Varilly, P., Shamovsky, O., Palma, A., Mikkelsen, T., Altshuler, D., and Lander, E. (2006). Positive Natural Selection in the Human Lineage. Science *312*, 1614–1620.

Ota, T., Suzuki, Y., Nishikawa, T., Otsuki, T., Sugiyama, T., Irie, R., Wakamatsu, A., Hayashi, K., Sato, H., Nagai, K., et al. (2004). Complete sequencing and characterization of 21,243 full-length human cDNAs. Nature Genetics *36*, 40–45.

Pruitt, K., Tatusova, T., and Maglott, D. (2007). NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. Nucleic Acids Research *35*, D61.

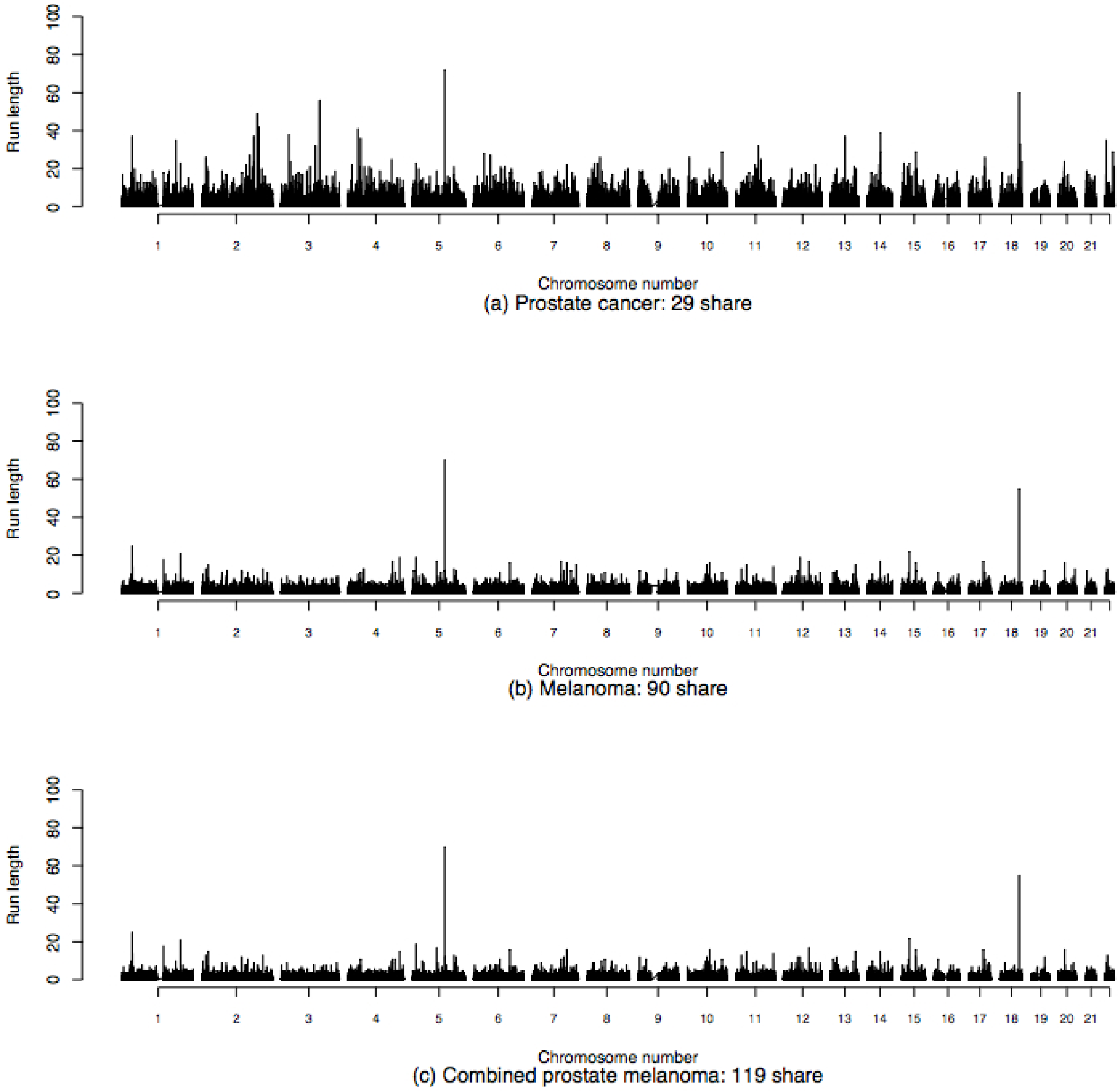
Haitina, T., Lindblom, J., Renström, T., and Fredriksson, R. (2006). Fourteen novel human members of mitochondrial solute carrier family 25 (SLC25) widely expressed in the central nervous system. Genomics *88*, 779–790.

Evans, P., Gilbert, S., Mekel-Bobrov, N., Vallender, E., Anderson, J., Vaez-Azizi, L., Tishkoff, S., Hudson, R., and Lahn, B. (2005). Microcephalin, a Gene Regulating Brain Size, Continues to Evolve Adaptively in Humans.

Mekel-Bobrov, N., Gilbert, S., Evans, P., Vallender, E., Anderson, J., Hudson, R., Tishkoff, S., and Lahn, B. (2005). Ongoing Adaptive Evolution of ASPM, a Brain Size Determinant in Homo sapiens.

This manuscript was prepared with the AAS LATEX macros v5.2.

6. Figures and Tables

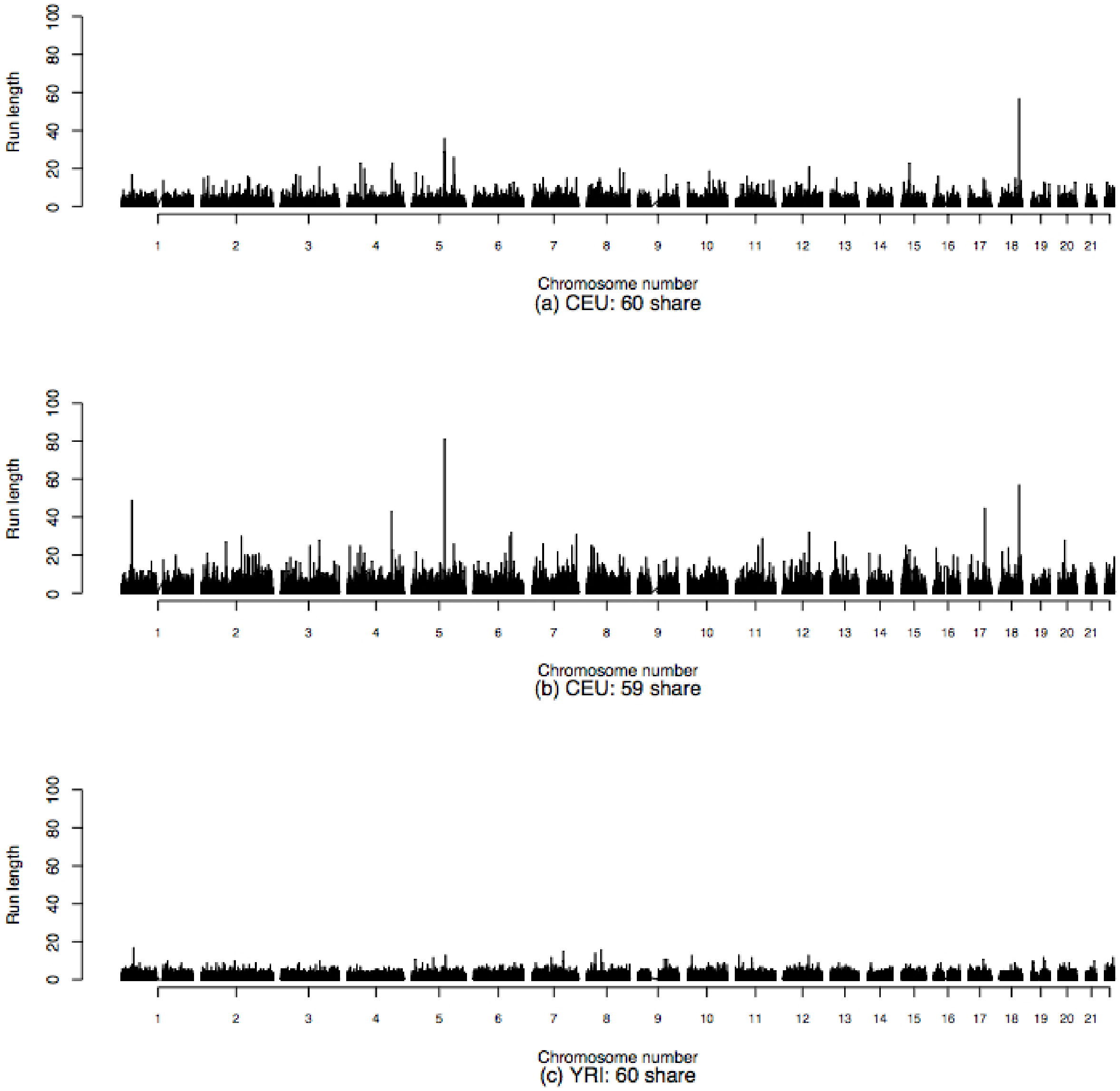


(a) Prostate cancer: 29 share

(b) Melanoma: 90 share

(c) Combined prostate melanoma: 119 share

Fig. 1.— Familial disease clusters

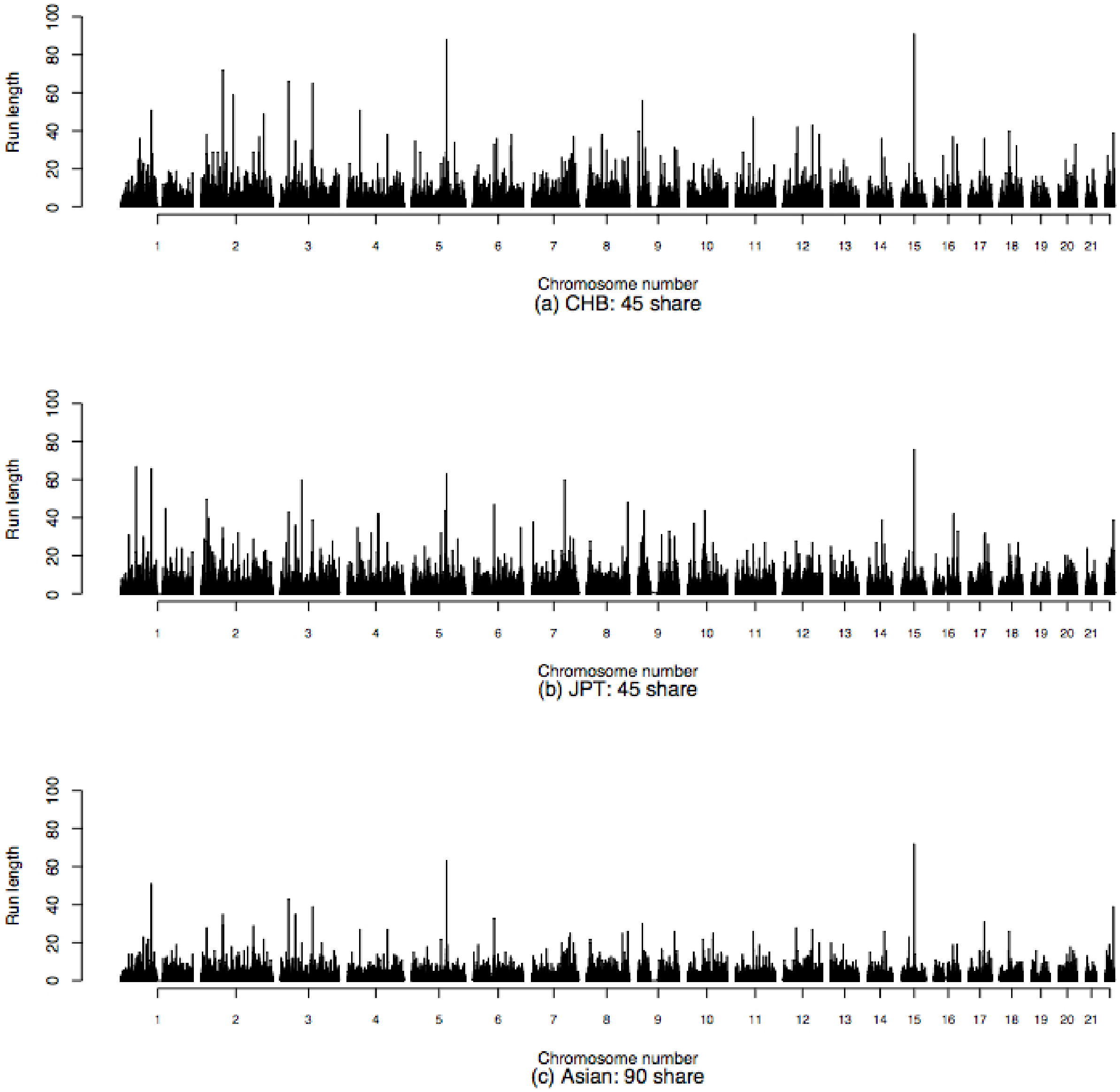


(a) CEU: 60 share

(b) CEU: 59 share

(c) YRI: 60 share

Fig. 2.— HapMap CEU and YRI data



(a) CHB: 45 share

(b) JPT: 45 share

(c) Asian: 90 share

Fig. 3.— HapMap CHB and JPT data

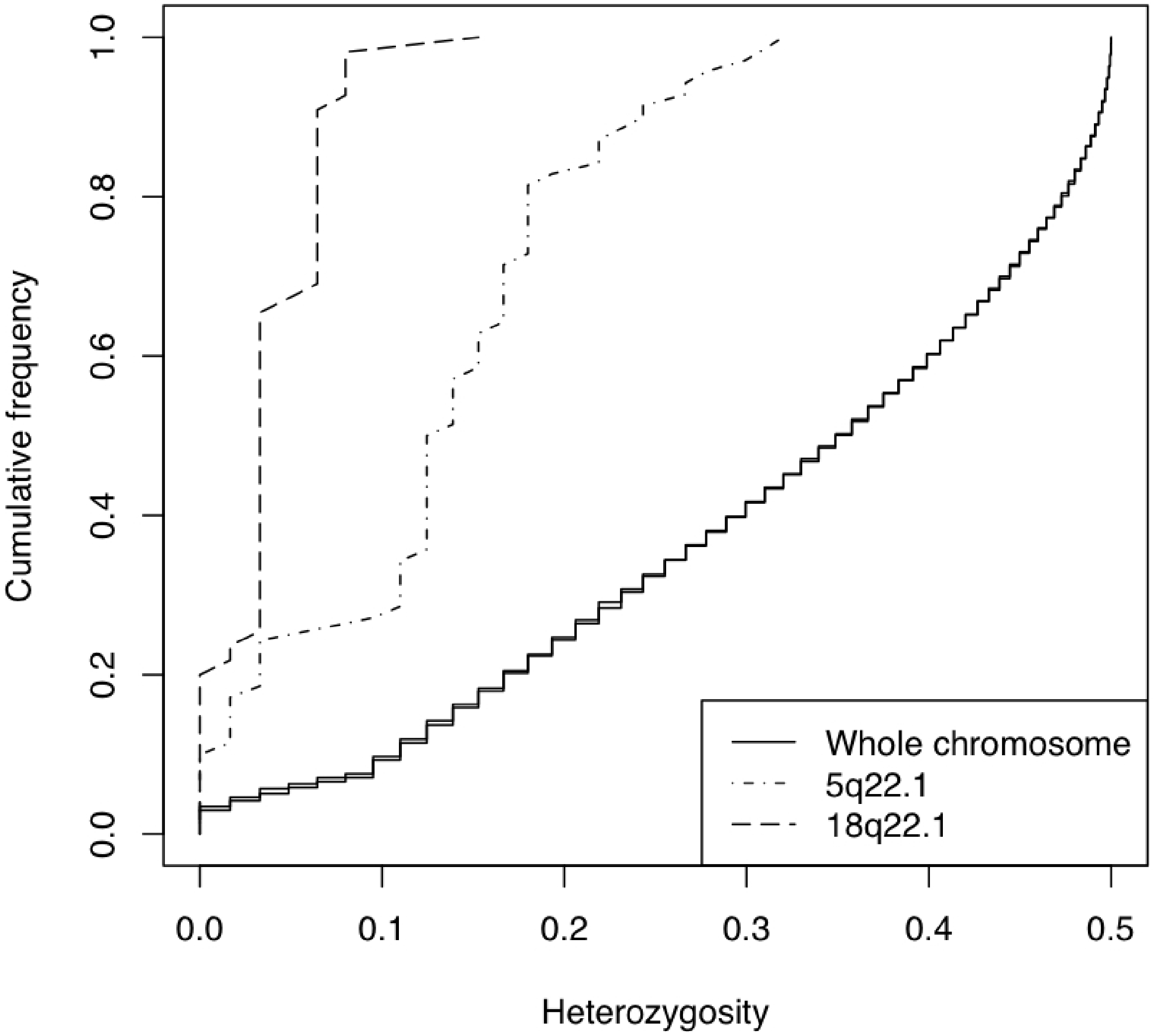


Fig. 4.— Heterozygosity analysis

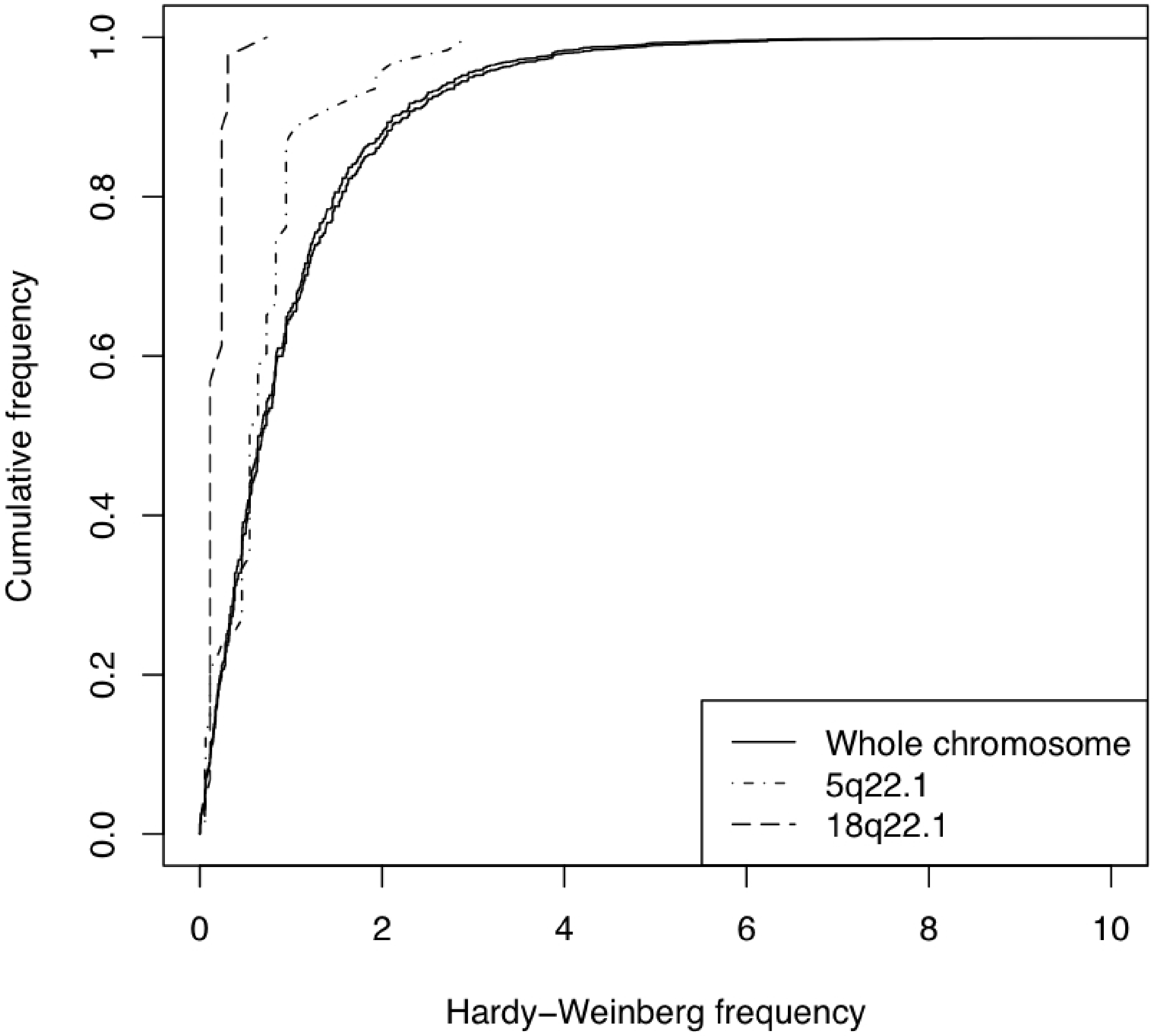


Fig. 5.— Hardy-Weinberg analysis

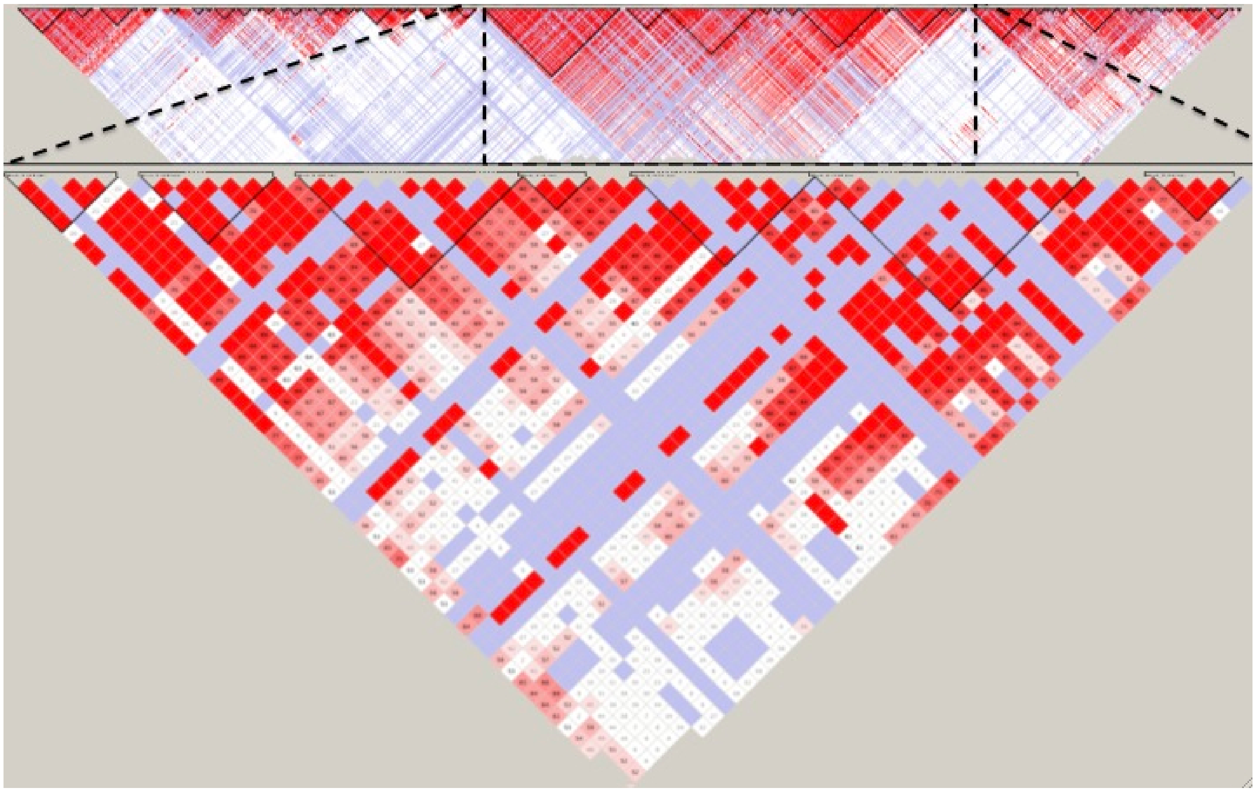


Fig. 6.— Linkage disequilibrium in the chromosome 5 region

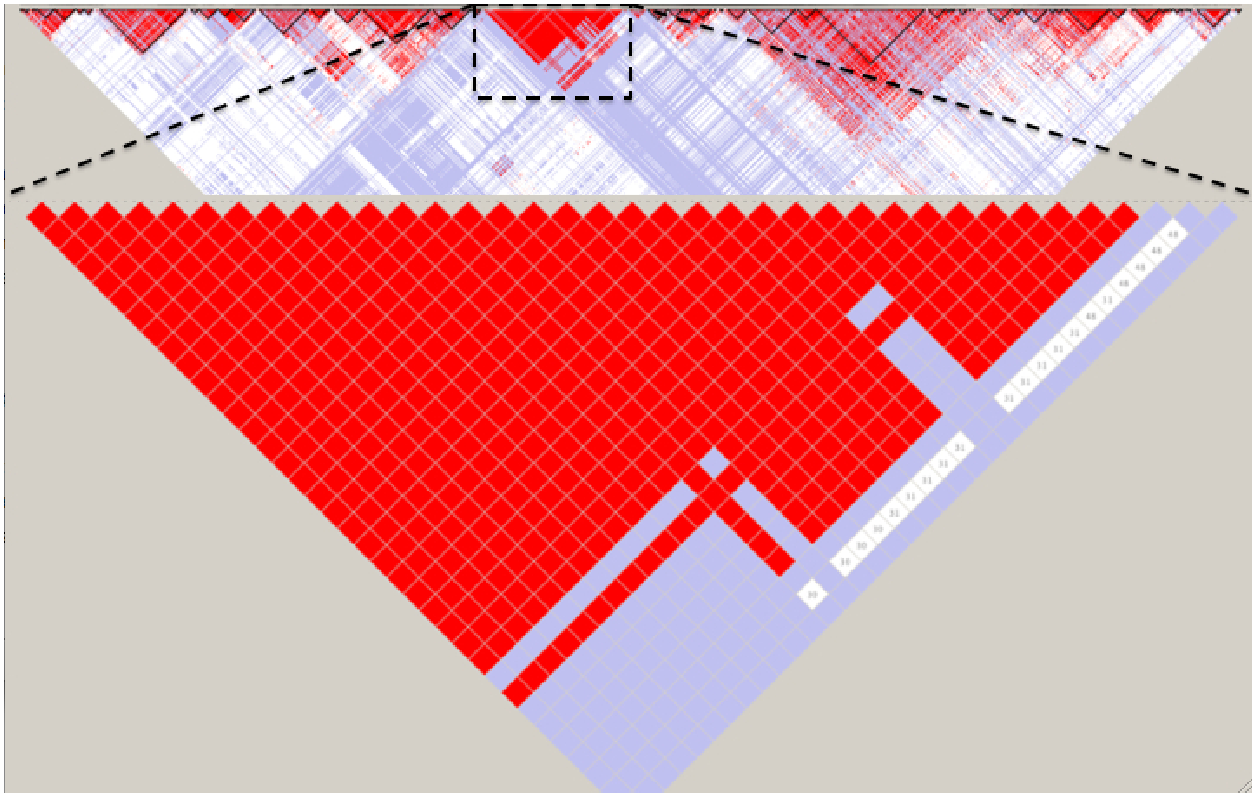


Fig. 7.— Linkage disequilibrium in the chromosome 18 region